Mesure et Analyse du Trafic de Réseaux : Théorie et Pratiques

Sandrine Vaton

Université de Rennes 1

École Doctorale MATISSE

MAthématiques, Télécommunications, Informatique,

Signal, et Système Electronique

HABILITATION À DIRIGER DES RECHERCHES

Présentée par

Mme Sandrine VATON

Mesure et Analyse du Trafic de Réseaux : Théorie et Pratiques

préparée à Télécom Bretagne, Brest

soutenue le 13 juillet 2012

Composition du jury :

M. Paulo Goncalves,	Chargé de Recherches, Ecole Normale Supérieure de Lyon,	Rapporteur
M. Philippe Owezarski,	Chargé de Recherches, LAAS/CNRS Toulouse,	Rapporteur
M. Patrick Thiran,	Professeur, Ecole Polytechnique Fédérale de Lausanne	Rapporteur
M. Guy Leduc,	Professeur des Universités, Université de Liège,	Examinateur
M. Gerardo Rubino,	Directeur de Recherches, INRIA Rennes - Bretagne Atlantique,	Examinateur
M. Bruno Tuffin,	Chargé de Recherches, INRIA Rennes - Bretagne Atlantique,	Examinateur
M. Jean Walrand,	Professeur des Universités, Université de Californie à Berkeley,	Examinateur
Mme Annie Gravey,	Directrice de Recherches, Télécom Bretagne,	Membre Invitée

Remerciements

Je remercie tout d'abord ma famille et mes collègues pour leurs encouragements à écrire ce manuscrit d'Habilitation à Diriger les Recherches. Le soutien de ma famille et les relances de mes collègues, m'ont poussée à me mettre et me remettre à la rédaction. Les urgences du quotidien auraient pu prendre le pas sur cet effort de synthèse, et ce document ne serait peut-être jamais sorti sans leur soutien.

Je remercie également l'ensemble des collègues de Télécom Bretagne, personnel administratif et enseignants-chercheurs, pour l'environnement de travail sympathique dont je bénéficie depuis plusieurs années. C'est toujours un plaisir de venir travailler le matin, dans un cadre privilégié.

Ma gratitude va vers les collègues, français et étrangers, académiques et industriels, avec lesquels je collabore depuis de nombreuses années. Les projets de recherche collaboratifs, les échanges académiques, m'ont permis de m'enrichir au contact d'autres chercheurs desquels j'ai beaucoup appris. Nos projets ont été l'occasion de poser un certain nombre de défis, qui ont contribué à faire mûrir et à structurer mon activité de recherche.

Mes sincères remerciements vont également aux jeunes chercheurs avec lesquels j'ai travaillé : doctorants, post-doctorants, stagiaires... Le travail présenté dans le cadre de ce manuscrit est le résultat de collaborations fructueuses avec ces jeunes, dont certains sont maintenant devenus des collègues. Je ne les remercierai jamais assez pour un enrichissement que j'espère mutuel.

Pour finir j'aimerais vivement remercier les membres du jury, et en particulier les rapporteurs avant soutenance, pour le temps qu'ils ont consacré à l'étude de mes travaux. C'est un honneur qu'ils me font en acceptant de participer au jury et c'est un grand plaisir que de voir mes travaux évalués par ces collègues que j'estime.

Structure du Manuscrit

CURRICULUM VITAE

Sandrine VATON
Née le 15 avril 1970
Mariée, 3 enfants
`http://perso.telecom-bretagne.eu/sandrinevaton/`

1 Biographie

Je suis titulaire du diplôme d'ingénieurs de TELECOM Paris et du DEA de Probabilités de l'Université Paris 6. Je suis également titulaire d'une thèse en traitement du signal de TELECOM Paris pour des travaux intitulés "Modélisation Statistique de Trafic dans les Réseaux". Je suis maître de conférences à TELECOM Bretagne depuis 1999. Je suis membre de l'IRISA depuis 2012.

Je monte, assure et coordonne des enseignements dans les domaines de l'évaluation des performances, de la sécurité des réseaux, du traitement du signal et des statistiques. J'interviens dans le cadre de la formation des ingénieurs Télécom Bretagne et de la Formation Continue. J'interviens également comme professeur invité dans des universités étrangères pour assurer des cours de niveau mastère recherche ou doctorat (Université de Pise, Université de la République d'Uruguay). J'organise régulièrement des cours pour doctorants dans le cadre de projets de recherche de la Commission Européenne (FP7 NoE EuroNGI, FP7 IP ETICS).

J'ai développé l'essentiel de mes activités de recherche dans le domaine de la mesure et de l'analyse du trafic dans les réseaux de télécommunications. Je suis l'auteur d'une cinquantaine de publications dans des revues internationales, des livres, des conférences internationales avec comité de lecture. J'ai été l'encadrant principal de 5 thèses de doctorat (3 thèses soutenues, 2 thèses en cours) ; j'ai également encadré trois post-doctorats ainsi que de très nombreux stages et projets d'étudiants.

Je monte et réalise des contrats de recherche en partenariat avec des partenaires académiques et industriels, dans le cadre de programmes de recherche de la Commission Européenne (FP7 NoE EuroNGI, EuroFGI et EuroNF, FP7 IP ETICS, FP7 IP DEMONS), de l'Agence Nationale de la Recherche (ANR OSCAR, ANR VIPEER), du Ministère des Affaires Etrangères (programmes Ecos-Sud, STIC AmSud), de l'Institut Télécom (programme Futur & Ruptures) ou en relation bilatérale avec l'industrie (thèse CIFRE). J'expertise des projets de recherche pour l'Institut Télécom (programme Futur & Ruptures) et pour l'Agence Nationale de la Recherche (programme VERSO).

J'organise annuellement le workshop TRAC "International Workshop on Traffic Analysis and Classification" (1ère édition à Caen en 2010, 2nde édition à Istanbul en 2011, 3ème édition à Chypre en 2012). Je fais partie du comité de programme de différentes conférences ou workshops (NGI, ISIVC, TRAC, ETICS 2012, MMRP12...). J'évalue des articles pour des revues internationales (Annales des Télécoms, European Transactions on

Telecommunications, Journal of Computer Security, Computers and Security, Neurocomputing, IEEE Journal on Selected Areas in Communication, Computer Networks, IEEE Communication Letters...).

2 Parcours Universitaire

- **1998 Diplôme de Doctorat en Traitement du Signal de TELECOM Paris**

  Titre : *Modélisation Statistique de Trafic dans les Réseaux*

  Thèse de doctorat de l'ENST ("Télécom Paris")

  Spécialité : Signal et Images

  Thèse soutenue le 11 décembre 1998

  Composition du Jury : Alain Hillion (président), Patrick Flandrin (rapporteur), Alain Simonian (rapporteur), Hayri Korezlioglu (examinateur), Marc Lavielle (examinateur), Eric Moulines (directeur de thèse)

- 1995 Diplôme d'Etudes Avancées en Probabilités de l'Université Paris 6
- **1994 Diplôme d'Ingénieur de TELECOM Paris**
- 1991 Concours Commun Mines-Ponts-Telecom ; admission à TELECOM Paris
- 1988-1991 Classes Préparatoires aux Grandes Ecoles (lycée Kléber, Strasbourg)
- 1988 Baccalauréat scientifique, mention très bien (lycée Courbet, Belfort)

3 Parcours Professionnel

- **1994-1995 Stages Ingénieur de Recherche à la Société Générale, Direction des Marchés de Capitaux, Produits Dérivés**
  - Développement de modèles mathématiques (en support à l'activité de trading)
  - Développement d'un logiciel pour la quotation de produits dérivés
  - Méthode des moments pour la quotation de produits dérivés.
- **1995-1998 Thèse de Doctorat à TELECOM Paris**
  - Département Traitement du Signal et des Images de TELECOM Paris, directeur de thèse : Prof. Eric Moulines
  - Contrat de Recherche Externalisé (CRE) avec France Télécom R&D (Lannion, responsable : Fabrice Clérot)
  - Mots Clés : Modélisation de Trafic ; Processus AutoSimilaires, A Dépendance Longue, Lois A Queues Lourdes, Processus Markoviens Quasi-Décomposables, Tests de Stationnarité, Estimation Paramétrique.
- **1995-1998 Vacations d'enseignement à TELECOM Bretagne**
  - Encadrement de TD en Probabilités, Processus Aléatoires, Traitement du Signal ; étudiants de 1ère et 2ème année du cycle ingénieurs à Télécom Bretagne
- **Depuis 1999 Maître de Conférences à TELECOM Bretagne**

  - 1999-2002 Département Electronique ; équipe de recherche en communications numériques (Turbo-Codes)
  - 2003-aujourd'hui Département Informatique ; équipe de recherche en métrologie et ingénierie du trafic
- **2000-2003 Vacations d'Enseignement à l'ENSAI**
  - Ecole Nationale de la Statistique et de l'Analyse de l'Information (Rennes) ; cours en 2ème année et en 3ème année filières Industrie et Santé
  - Méthodes de Monte Carlo par Chaînes de Markov (MCMC)
- **Séjours d'études dans des Universités Etrangères**
  - Plusieurs séjours de quelques semaines à l'***Université Technologique d'Helsinki*** (HUT, Finlande) ; équipe du professeur Jorma Virtamo
  - Plusieurs séjours d'une semaine à l'***Université de Pise*** (Italie) ; équipe du professeur Stefano Giordano
  - Septembre 2010 à janvier 2011 : séjour d'études à l'***Institut d'Ingénierie Electrique de l'Université de la République*** (UdelaR, Uruguay) ; groupe ARTES, équipe du professeur Pablo Belzarena
- **Membre de l'IRISA**
  - département D2 "Réseaux Télécom et Services", équipe REOP "Réseaux Opérés"
  - intégration IRISA en 2012 d'une vingtaine d'enseignants-chercheurs Télécom Bretagne

4 Contrats de Recherche et Financements

- **Contrat de Recherche Externalisé France Télécom R&D** sur la Modélisation Statistique de Trafic, 1991-1994, TELECOM Paris, Financement de la thèse de doctorat.
- **Projet ECOS-Sud "Internet as a convergence network"** avec IIE/UdelaR (Uruguay), INRIA et l'université de Paris-Sud, 2003-2005
- **SATURNE II, projet du Groupe des Ecoles des Télécommunications** (GET, maintenant Institut Mines Télécom), 2004. Monitoring actif de Qualité de Service.
- **Contrat d'encadrement de thèse** de Hamza DAHMOUNI (CIFRE avec Orange Labs, Belfort), 2004-2007
- **DYNAM'IT, projet du Groupe des Ecoles des Télécommunications**, 2006, Coordinatrice du projet. Ingénierie robuste et dynamique de trafic.
- **Réseaux d'Excellence Européens Euro-NGI, Euro-FGI, et Euro-NF**. Site Web public : `http://euronf.enst.fr/en_accueil.html`.
  Coordination d'une activité de recherche sur la matrice de trafic (HUT Helsinki, KTH Stockolm, Univ. Roma 2 et Roma 3, France Télécom, Institut Télécom) ; Organisatrice d'un Cours Doctoral de 2 semaines à Brest en 2006.
- **OSCAR, "Overlay Networks Security : Characterization, Analysis and Recovery"**, projet pré-compétitif de l'Agence Nationale de la Recherche (ANR-RNRT), 2006-2008.
  Détection d'anomalies de trafic dans le réseau d'opérateur. Développement d'algotithmes de détection, mise en place d'une plateforme de démonstration distribuée entre les partenaires.
- **VIPEER, "Video Traffic Engineering in an IntraDomain Context based on the P2P Paradigm"**, projet de l'Agence Nationale de la Recherche (ANR, pro-

gramme VERSO), 2010-2013. Site Web public : `http://recherche.telecom-bretagne.eu/vipeer/`.
Travaux sur la prise en compte de la métrologie dans la gestion des contenus video en streaming.

- **ETICS, "Economics and Technologies for Inter-Carrier Services"**, Projet Intégré du 7ème PCRD de la Commission Européenne (FP7, ICT, Call 4), 2010-2012. Site Web public : `https://www.ict-etics.eu/`.
Conception, développement, protorypage de l'architecture de mesure de SLA. Organisation du PhD Course `http://conferences.telecom-bretagne.eu/phdcourse`. *Travaux sur la vente, par des mécanismes d'enchère, de services avec QoS garantie en interdomaine . Travaux sur la répartition entre domaines des revenus générés par la vente de services garantis. Travaux sur le calcul robuste de paramètres de QoS en présence d'incertitude sur la demande de trafic.*
- **DEMONS, "Decentralized Monitoring for Trustworthiness"**, Projet Intégré du 7ème PCRD de la Commission Européenne (FP7, ICT, Call 5), 2010-2013. Site Web public : `http://fp7-demons.eu/`.
Développement d'un point de capture programmable, flexible et haute performance (BlockMon), en collaboration avec certains partenaires du projet (NEC Heidelberg, ETH Zürich, Uni. Pisa, INVEA-TECH, Institut Télécom). Accélération Matérielle de la Surveillance de Trafic (cartes NetFPGA et COMBOv2). Coordination du scénario sur la détection de dénis de service distribués (DDoS).
- **MAITRE**, projet du Ministère des Affaires Etrangères et Européennes, programme STIC AmSud, collaboration entre la France (Institut Mines Télécom, LAAS/CNRS), l'Argentine (UBA) et l'Uruguay (UdelaR), 2011-2012. Coordinatrice internationale du projet.
- **Plateforme Haute Performance/Haut Débit pour la Surveillance de Trafic**, 2011, Financement Institut Télécom (programme "Grands Projets de Rupture").

5 Activités d'Organisation, Expertise, Responsabilités Locales

5.1 Participation à des jurys de thèse

- Jury de thèse de **Pascal CHEUNG MON CHAN** (2003, TELECOM Paris). *Réseaux bayésiens et filtres particulaires pour l'égalisation adaptative et le décodage conjoints.* Participation en tant que membre invité.
- Jury de thèse de **Hamza DAHMOUNI** (mars 2007, Télécom Bretagne). *Reconnaissance de services, caractérisation de trafic et évaluation de performances dans les réseaux mobiles multi-services.* Participation en tant qu'examinatrice (encadrante de la thèse).
- Jury de thèse de **Dmitry MARAKOV** (décembre 2007, Télécom Bretagne). *Contributions to the traffic matrix problem.* Participation en tant qu'examinatrice (encadrante de la thèse).
- Jury de thèse de **Pedro CASAS** (juillet 2010, Télécom Bretagne). *Analyse Statistique du Trafic Réseau pour la Détection de Menaces de Sécurité et la Qualité de Service.* Participation en tant qu'examinatrice (encadrante de la thèse).
- Jury de thèse de **Silvio VALENTI** (septembre 2011, TELECOM ParisTech). *Dealing with P2P Traffic in Modern Networks : Measurement, Identification and Control.* Participation en tant qu'examinatrice.

– Jury de thèse de **Johan MAZEL** (décembre 2011, LAAS/CNRS). *Détection non-supervisée d'anomalies dans le trafic réseau.* Participation en tant qu'examinatrice.

5.2 Expertise de projets de recherche

– Expertises pour l'**Institut Télécom** ; programmes Futur & Ruptures 2010 et 2011
– Expertises pour l'**Agence Nationale de la Recherche** ; programme VERSO

5.3 Organisation de workshops et conférences

– **Chairman de TRAC, International Workshop on Traffic Analysis and Classification** (avec Christian Callegari, université de Pise).
  – workshop organisé dans le cadre de la conférence IWCMC
  – TRAC 2010, Caen, juillet 2010, **(Best IWCMC 2010 Workshop Award)**
  – TRAC 2011, Istanbul, juillet 2011, **(Best IWCMC 2011 Workshop Award)**
  – TRAC 2012, Limassol, Chypre, août 2012
– **Membre du TPC de ISIVC, Fitramen 2008, NGI, TRAC, ETICS 2012, MMRP 2012**
– Membre du comité d'organisation du 2nd International Symposium on Turbo Codes & Related Topics, Brest, France, 2000.

5.4 Evaluations d'articles soumis en journaux

Expertises pour différents journaux : IEEE Transactions on Information Theory, IEEE Journal on Selected Areas in Communication, IEEE Communication Letters, Computer Networks (Elsevier), Annales des Télécommunications (Institut Télécom, Springer), European Transactions on Telecommunications (Wiley), Journal of Computer Security (IOS Press), Computers and Security (Elsevier), Neurocomputing (Elsevier), Telecommunications System (Springer), Journal of Communications and Networks (Korea Information and Communication Society)...

5.5 Responsabilités Locales

– Représentante des enseignants-chercheurs de TELECOM Bretagne au sein du **Réseau Thématique RT2 "Architecture et Gestion de Réseaux"** de l'Institut Télécom.
– Membre élu du personnel représentant le collège maître de conférences à la **Commission des Appelations** de TELECOM Bretagne.
– Responsable scientifique à TELECOM Bretagne des **échanges d'étudiants en double diplôme avec la Pologne** (en particulier GUT, Gdansk University of Technoloy).

6 Prix, Distinctions

– **Best IWCMC workshop award** pour l'organisation de TRAC 2010 (avec C.Callegari de l'Université de Pise)
– **Best IWCMC workshop award** pour l'organisation de TRAC 2011 (avec C.Callegari de l'Université de Pise)

- **Best LANOMS'09 Conference Paper Award** pour l'article "Robust routing mechanisms for intradomain traffic engineering in dynamic networks" par CASAS HERNANDEZ Pedro, LARROCA Federico, VATON Sandrine
- **Best NGI 2008 Conference Paper Award** pour l'article "Robust and reactive traffic engineering for dynamic traffic demands" par CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine.

7 Encadrements de thèses et de post-doctorats

7.1 Thèses soutenues

1. **Dr Horacio GONZALEZ**
   - *Nouveaux schémas de turbo-codage*
   - Thèse encadrée par Claude Berrou. Co-encadrement de thèse durant la première année.
   - Thèse soutenue en décembre 2004
   - Poste actuel : ingénieur système, Crédit Mutuel Arkéa
2. **Dr Hamza DAHMOUNI**
   - *Reconnaissance de services, caractérisation de trafic et évaluation de performances dans les réseaux mobiles multi-services*
   - Thèse CIFRE avec Orange Labs (Belfort)
   - Thèse soutenue en mars 2007
   - Résumé : `http://departements.telecom-bretagne.eu/info/theses/theses_soutenues/dahmouni_hamza.php`
   - Poste actuel : maître de conférences, INPT Rabat
3. **Dr Dmitry MARAKOV**
   - *Contributions to the traffic matrix problem*
   - Thèse soutenue en décembre 2007
   - Résumé : `http://departements.telecom-bretagne.eu/info/theses/theses_soutenues/marakov_dimitri.php`
   - Poste actuel : ingénieur chez Saooti (web radio, Lannion)
4. **Dr Pedro CASAS**
   - *Analyse Statistique du Trafic Réseau pour la Détection de Menaces de Sécurité et la Qualité de Service*
   - Thèse soutenue en juillet 2010
   - Résumé : `http://departements.telecom-bretagne.eu/info/theses/theses_soutenues/casas_pedro.php`
   - Poste actuel : chercheur senior à FTW (Vienne, Autriche)

7.2 Thèses en cours

1. **Mlle Isabel AMIGO**
   - *Métrologie, Ingénierie du Trafic et Modèles Economiques pour la QoS de bout-en-bout en Inter-Domaine*
   - Début de thèse : mars 2010
   - Thèse co-encadrée par Pablo Belzarena (IIE, UdelaR, Uruguay)
   - Résumé : `http://departements.telecom-bretagne.eu/info/theses/theses_en_cours/amigo_maria.php`

2. **M. Tristan GROLEAT**
   - *Accélération Matérielle pour la Surveillance de Trafic à Très Haut Débit*
   - Début de thèse : janvier 2011
   - Thèse co-encadrée par Matthieu Arzel (TELECOM Bretagne, Electronique)

7.3 Post-doctorants

1. **Dr Lionel FILLATRE**
   - Détection et Localisation d'Anomalies dans la Demande de Trafic sur un Système Autonome par des Approches Non Invasives
   - Post-doctorat à TELECOM Bretagne en 2006
   - Page personnelle : `http://fillatre.pagesperso-orange.fr/`
   - Poste actuel : maître de conférences, UTT, Troyes
2. **Dr Osman SALEM**
   - Détection de Comportements Malicieux par une Surveillance Statistique du Trafic dans un Réseau d'Opérateur : Algorithmes et Expérimentations
   - Post-doctorat à TELECOM Bretagne en 2007-2008
   - Post-doctorat dans le cadre du projet pré-compétitif ANR-RNRT OSCAR "Overlay Networks Security : Characterization, Analysis and Recovery"
   - Page personnelle : `http://www.mi.parisdescartes.fr/~osalem/`
   - Poste actuel : maître de conférences à l'université Paris 7
3. **Dr Mohamed Karim SBAI**
   - Surveillance du Trafic pour le Contrôle d'un Réseau de Distribution de Contenus Vidéo Opéré par le Fournisseur d'Accès
   - Post-doctorat à TELECOM Bretagne d'octobre 2010 à novembre 2012
   - Post-doctorat dans le cadre des projets ANR VIPEER (programme VERSO) et FP7 IP DEMONS

7.4 Accueil de chercheurs associés à Télécom Bretagne

Accueil de deux chercheurs associés au département Informatique de Télécom Bretagne
- **Dr Hamza DAHMOUNI**, maître de conférences INPT Rabat, modélisation probabiliste de réseaux et analyse du trafic
- **Dr Serge GUELTON**, ingénieur de recherche, compilation et calcul haute performance

7.5 Stages de niveau ingénieur ou mastère

Nombreux encadrements de stages d'étudiants français et étrangers dans le cadre de leur formation d'ingénieur (ou de niveau mastère).

8 Accueil de chercheurs étrangers

8.1 Accueil de doctorants étrangers

Dans le cadre de collaborations internationales j'ai accueilli pendant quelques mois des doctorants étrangers à TELECOM Bretagne.

1. **Dr Christian CALLEGARI**

- Thèse sur la détection d'anomalies dans le trafic de réseau, Université de Pise, TLC Networks Research Group, encadrant : Michele PAGANO
- 2006-2007 : deux séjours de recherche de quelques mois à TELECOM Bretagne
- Sujet : détection d'anomalies TCP avec des modèles de chaînes de Markov d'ordre élevé
- Page personnelle : `http://netserv.iet.unipi.it/people/callegari/Home_Page.html`

2. **Dr Ilmari JUVA**
   - Thèse sur la matrice de trafic, HUT Helsinki, encadrant : Jorma VIRTAMO
   - 2006 : séjour de recherche d'un mois à TELECOM Bretagne
   - Sujet : méthodes légères pour l'estimation de la matrice de trafic
   - Lien : `http://lib.tkk.fi/Diss/2008/isbn9789512292998/`
3. **M. Jan KASTIL**
   - Thèse à l'université technologique de Brno, République Tchèque
   - 2010 : séjour de recherche à TELECOM Bretagne
   - Sujet : classification de trafic par des méthodes comportementales
4. **Mlle Erin DOOLITTLE**
   - Thèse à l'université technologique de Clemson, Caroline du Sud
   - juillet 2010 : séjour de recherche à TELECOM Bretagne
   - Sujet : optimisation robuste multi-critères

8.2 Accueil de chercheurs seniors

J'ai accueilli des chercheurs étrangers pour des séjours de quelques semaines à TELECOM Bretagne dans le cadre de divers projets collaboratifs :

1. **Dr. Michele PAGANO**, TLC Networks Research Group, Université de Pise ; collaboration dans le cadre du réseau d'excellence EuroNGI
2. **Dr. Natalia MARKOVICH**, chercheur, Institut des Sciences du Contrôle, Académie des Sciences, Moscou, collaboration dans le cadre du réseau d'excellence EuroNGI
3. **Prof. Gonzalo PERERA**, Professeur de Grade 5, Laboratoire de Probabilités et Statistiques, Institut d'Ingéniérie, Université de la République, Uruguay, collaboration dans le cadre du projet *Internet comme réseau de convergence* (MAEE, programme ECOS-Sud)
4. **Prof. Pablo BELZARENA**, Professeur de Grade 5, responsable de l'Institut d'Ingénierie Electrique, Université de la République, Uruguay, collaboration dans le cadre des projets *Internet comme réseau de convergence* (MAEE, programme ECOS-Sud) et MAITRE (MAEE, programme STIC AmSud)
5. **Dr Hervé KERIVIN**, Mathematical Sciences, Clemson University, Caroline du Sud, USA, séjours d'un mois à TELECOM Bretagne en juillet 2009 et juillet 2010 suite à une collaboration démarrée dans le projet DYNAM'IT (Institut Télécom, programme Futur & Ruptures)

9 Enseignements à TELECOM Bretagne

9.1 Théorie des Files d'Attente

- 2004 Mise en place d'un cours de théorie des files d'attente à TELECOM Bretagne

- Responsable du cours depuis 2004
- Cours de "Majeure Réseaux" (1ère et 2ème années, Formation d'Ingénieurs)
- Cours de 3ème année de Formation d'Ingénieurs en Partenariat
- 63 heures d'enseignement/an à TELECOM Bretagne
- Mots-clés : chaînes de Markov (temps discret, temps continu), files markoviennes (M/M/1, M/M/C/C, ...), files généralisées (M/G/1, G/M/1), réseaux de Jackson, travaux pratiques, exercices.
- Cours de Formation Continue "Files d'Attente et Modélisation de Trafic"

9.2 Sécurité Informatique

- 2003 Mise en place d'un cours de cryptographie, TELECOM Bretagne, 3ème année Filière "Systèmes Logiciels et Réseaux" sur Brest
- Depuis 2008 Responsable du cours de Sécurité Informatique, TELECOM Bretagne, 3ème année Filière "Systèmes Logiciels et Réseaux" sur Brest, 61 heures/an
- Mots-clés : cryptographie (clé publique, cryptographie symétrique, preuves sans transfert de connaissance), authentification (Kerberos), protocoles de communication sécurisés (SSH, IPSec, ...), politiques de sécurité, watermaking, biométrie, attaques matérielles, Filtrage, Détection d'Intrusions, Obfuscation de Code, Travaux Pratiques (craquage de mots de passe, certification de clés, VPN, IDS, watermarking, ...)

9.3 Méthodes Avancées en Statistiques

- 2004 Mise en place d'un cours sur des méthodes statistiques basées sur la simulation informatique à TELECOM Bretagne
- 2004-2008 Responsable de ce cours dans la "Majeure Mathématiques et Traitement du Signal" (1ère et 2ème année, Formation d'Ingénieurs)
- 42 heures d'enseignement/an à TELECOM Bretagne
- Mots clés : théorèmes limites, méthodes de Monte Carlo, méthodes MCMC (échantillonage d'importance, acceptation/rejet, Hastings Metropolis, Gibbs, recuit simulé,...), algorithme EM, théorie des grandes déviations

9.4 Autres cours

Avant la dernière réforme de l'enseignement (2004) j'assurais des cours dans différents domaines :

- Communications numériques : codage correcteur d'erreurs (formation d'ingénieurs et formation continue)
- Théorie de l'information
- Probabilités, statistiques, processus stochastiques
- Traitement du signal : techniques de filtrage (Kalman, particulaire, ...)
- Algorithmique
- Programmation en langage C
- Electronique numérique

9.5 Encadrement de Projets

Les projets sont une composante importante de la formation à TELECOM Bretagne. Je propose et je co-encadre avec mes collègues, mes doctorants et post-doctorants, des projets d'étudiants variés à TELECOM Bretagne. Exemples de sujets :

- 2011-2012
  1. Génération de Trafic avec NetFPGA (3ème année),
  2. Détection de trafic P2P et reconfiguration de firewall dans le réseau des élèves (1ère année),
  3. Apprentissage SVM sur carte graphique (3ème année),
  4. Accélération matérielle sur FPGA de méthodes de Monte Carlo pour la quotation de produits financiers dérivés (2ème année)
- 2009-2010 Détection de Flots Massifs sur NetFPGA (3ème année)
- 2008-2009
  1. Automatisation de la mesure de qualité d'expérience (QoE) en voix sur IP (2ème année)
  2. Mise en place d'une plateforme de capture de trafic (DAG) et de classification en temps réel des flots par des méthodes d'apprentissage statistique
- 2007 Modèles de Performance pour l'UMTS
- 2005 Topologie de l'Internet, Analyse de Données RPSL

10 Enseignement au Niveau Doctoral (PhD Courses)

1. **2005 – Conférencière invitée. Institut d'Ingénierie Electrique (UdelaR, Uruguay).** Méthodes Statistiques Avancées pour l'Analyse du Trafic, 15 heures de cours.
2. **Juin 2006 – Organisation d'un PhD Course "IP traffic characterization, data analysis and statistical methods", TELECOM Bretagne,** 2 semaines en juin 2006, sponsor : réseau d'excellence européen EuroNGI
3. **Automne 2006 – Conférencière dans le cadre du PhD Course "IP traffic characterization, data analysis and statistical methods", Université de Pise,** automne 2006, sponsor : réseau d'excellence européen EuroNGI
4. **Juin 2010 – Conférencière pour un cours sur "Statistical Methods for the OD traffic matrix problem",** programme de formation doctorale de l'**Université de Pise**, Italie (20 heures de cours).
5. **Automne 2010 – Conférencière pour le cours "Statistical Methods for the OD traffic matrix problem" à l'Institut d'Ingénierie Electrique de l'UdelaR** (Universidad de la Republica, Uruguay) , (24 heures de cours et encadrement de projets d'étudiants).
6. **Octobre 2011 – Organisation d'un PhD Course "Economics and Technologies for InterCarrier Services"**, TELECOM Paris, 17 au 21 octobre 2011, sponsor : projet intégré européen FP7 ETICS, réseau d'excellence européen FP7 EuroNF.

11 Publications

Voir le chapitre correspondant.

12 Langues

- Français : langue maternelle
- Anglais : maîtrise parfaite en environnement professionnel
- Allemand : niveau intermédiaire
- Espagnol : niveau intermédiaire
- Polonais : niveau débutant

LISTE DE PUBLICATIONS

H-index = 10, nombre de citations = 428 (Google Scholar, mars 2012)

1 Articles en journaux

1.1 Journaux Internationaux avec Comité de Lecture

1. AMIGO Isabel, VATON Sandrine, CHONAVEL Thierry, LARROCA Federico, Maximum Delay Computation for Interdomain Path Selection, International Journal of Network Management (IJNM), volume 22, issue 2, pages 162-179, march/april 2012
2. CASAS HERNANDEZ Pedro, LARROCA Federico, ROUGIER Jean-Louis, VATON Sandrine, Taming traffic dynamics : analysis and improvements, Computer Communications (COMCOM), volume 35, issue 5, pages 565-578, march 2012
3. CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine, NIKIFOROV Igor, Reactive Robust Routing : Anomaly Localization and Routing Reconfiguration for Dynamic Networks, Journal of Network and Systems Management (JNSM), 2011, volume 19, issue 1, Pages 58-83
4. CALLEGARI Christian, VATON Sandrine, PAGANO Michele, A New Statistical Method for Detecting Network Anomalies in TCP Traffic, European Transactions on Telecommunications (ETT), volume 21, issue 7, november 2010
5. SALEM Osman, VATON Sandrine, GRAVEY Annie, A scalable, efficient and informative approach for anomaly-based Intrusion Detection Systems : theory and practice, International Journal of Network Management (IJNM), Special Issue : Traffic Monitoring and Network Measurements : from Theory to Practice, volume 20, issue 5, september/october 2010
6. CASAS HERNANDEZ Pedro, VATON Sandrine, FILLATRE Lionel, NIKIFOROV Igor, Optimal Volume Anomaly Detection and Isolation in Large-Scale IP Networks using Coarse-Grained Measurements, Computer Networks (COMNET), volume 54, issue 11, pages 1750-1766, august 2010, ISSN 1389-1286

1.2 Journaux Nationaux avec Comité de Lecture

1. DAHMOUNI Hamza, EN-NOUAARY A., VATON Sandrine, Modèle de chaîne de Markov pour la classification automatique des services, Revue Méditerrannéenne des Télécommunications (RMT), colume 1, numéro 1, janvier 2011

2 Chapitres de Livres

1. GLAVIEUX Alain, ADDE Patrick, BATTAIL Gérard, DOUILLARD Catherine, PICART Annie, PYNDIAH Ramesh, VATON Sandrine, POLI Alain, JEZEQUEL Michel, BIGLIERI Ezio, Channel coding in communication networks : from theory to turbocodes. Edited by Alain Glavieux. London : ISTE Ltd, 2007, 418 p. ISBN 978-1-905209-24-8
2. VATON Sandrine, BEDO J.S, GRAVEY Annie, Advanced Methods for the Estimation of the Origin Destination traffic matrix. Performance Evaluation and Planning Methods for the Next Generation Internet, Editors : A. Girard, B. Sanso, F. Vazquez-Abad, 2005

3. GLAVIEUX Alain, ADDE Patrick, BATTAIL Gérard, BIGLIERI Ezio, JEZEQUEL Michel, POLI Alain, VATON Sandrine, PYNDIAH Ramesh, PICART Annie, DOUILLARD Catherine, BERROU Claude, Codage de canal - des bases théoriques aux turbocodes. Paris : Hermès Science / Lavoisier, 2005, 453 p. (Traité IC2 : Traitement du signal et de l'image), ISBN 2-7462-0953-5
4. VATON Sandrine, KOREZLIOGLU Hayri, CHONAVEL Thierry, Is the apparent self-similarity of the broadband traffic due to non stationarity. Publié dans Performance Analysis of ATM Networks (Editeur : Demetres Kouvatsos), Kluwer Academic Publishers, 1999.

3 Articles en conférences

3.1 Conférences/Workshops Internationaux avec Comité de Lecture

1. HUICI Felipe, DI PIETRO Andrea, BONELLI Nicola, TRAMMELL Brian, VATON Sandrine, GROLEAT Tristan, KASTOVSKY Petr, NUCCILLI Fabrizio, Blockmon : A Modular System for Flexible, High-Performance Traffic Monitoring and Analysis, Usenix Annual Technical Conference 2012 (USENIX ATC'12) (soumis)
2. GROLEAT Tristan, ARZEL Matthieu, VATON Sandrine, Hardware Acceleration of SVM-Based Traffic Classification on FPGA, 3rd International Workshop on Traffic Analysis and Classification (IWCMC TRAC 2012), Limassol, Chypre, août 2012
3. AMIGO Isabel, BELZARENA Pablo, VATON Sandrine, On the Problem of Revenue Sharing in Multi-Domain Federations, Networking 2012, Prague, mai 2012
4. AMIGO Isabel, BELZARENA Pablo, LARROCA Federico, VATON Sandrine, Network Bandwidth allocation with end-to-end QoS constraints and Revenue Sharing in Multidomain Federations, 7th International ICQT Workshop on Advanced Internet Charging and QoS Technology (ICQT'2011), Paris, octobre 2011
5. AMIGO Isabel, VATON Sandrine, CHONAVEL Thierry, LARROCA Federico, Maximum Delay Computation under Traffic Matrix Uncertainty and its Application to Interdomain Path Selection, International Network Optimization Conference (INOC'2011), Hamburg, juin 2011
6. N. Le Sauze, A. Chiosi, R. Douville, H. Pouyllau, H. Lonsethagen, P. Fantini, C. Palasciano, A. Cimmino, M. A. Callejo Rodriguez, O. Dugeon, D. Kofman, X. Gadefait, P. Cuer, N. Ciulli, G. Carrozzo, A. Soppera, B. Briscoe, F. Bornstaedt, M. Andreou, G. Stamoulis, C. Courcoubetis, P. Reichl, I. Gojmerac, J. L. Rougier, S. Vaton, D. Barth, A. Orda, ETICS : QoS enabled interconnection for future Internet services, Future Network and Mobile Summit, Pisa, Italie, juin 2010
7. CASAS HERNANDEZ Pedro, VATON Sandrine, On the use of random neural networks for traffic matrix estimation in large-scale IP networks, 1st International Workshop on Traffic Analysis and Classification (TRAC, co-located with IWCMC 2010), Caen, 28 juin-2 juillet 2010
8. CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine, NIKIFOROV Igor Volume anomaly detection in data networks : an optimal detection algorithm vs the PCA approach, Lecture Notes in Computer Science, 2009, vol. 5464, actes du workshop Fitramen 2008, Porto, décembre 2008
9. CASAS HERNANDEZ Pedro, VATON Sandrine, FILLATRE Lionel, CHONAVEL Thierry Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks. ITC'21 : 21st conference on International Teletraffic Congress, 15-17 september 2009, Paris, France, 2009
10. CASAS HERNANDEZ Pedro, LARROCA Federico, ROUGIER Jean Louis, VATON Sandrine Robust routing versus dynamic load balancing : a comprehensive study and new directions. DRCN'09 : 7th International Workshop on the Design of Reliable Communication Network, 25-28 october 2009, Washington, United States, 2009
11. CASAS HERNANDEZ Pedro, LARROCA Federico, VATON Sandrine Robust routing mechanisms for intradomain traffic engineering in dynamic networks. LANOMS'09 : 6th Latin-American Network Operations and Management Symposium, 19-21 september 2009, Punta Del Este, Uruguay, 2009. **Best Conference Paper Award**.

12. SALEM Osman, MEHAOUA Ahmed, VATON Sandrine, GRAVEY Annie Flooding Attacks Detection and Victim Identification over High Speed Networks. GIIS'2009 : IEEE Global Information Infrastructure Symposium, 22-26 june 2009, Hammamet, Tunisia, 2009
13. CALLEGARI Christian, VATON Sandrine, PAGANO Michele A new statistical approach to network anomaly detection. SPECTS'08 : International Symposium on Performance Evaluation of Computer and Telecommunication Systems, June 16-18, Edinburgh, UK, 2008
14. CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine Multi hour robust routing and fast load change detection. ICC 2008 : IEEE International Conference on Communications, may 19-23, Beijing, China,
15. CASAS HERNANDEZ Pedro, VATON Sandrine, FILLATRE Lionel, NIKIFOROV Igor Volume anomaly detection in data networks : an optimal volume detection vs the PCA approach. Fitra-Men'08 : International Workshop on Traffic Management and Traffic Engineering for the Future Internet, december 11-12, Porto, Portugal , 2008
16. CASAS HERNANDEZ Pedro, BELZARENA Pablo, VATON Sandrine End-2-End evaluation of IP multimedia services, a user-perceived QoS approach. 18th ITC Specialist Seminar on Quality of Experience, may 29-30, Karlskrona, Sweden, 2008
17. CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine Robust and reactive traffic engineering for dynamic traffic demands. NGI 2008 : 4th EuroNGI Conference on Next Generation Internet Networks, april 28-30, Krakow, Poland , 2008. **Best Conference Paper Award**.
18. FILLATRE Lionel, NIKIFOROV Igor, VATON Sandrine, CASAS HERNANDEZ Pedro Sequential non Bayesian network traffic flows anomaly detection and isolation. IWAP 2008 : International Workshop on Applied Probability, July 7-10, Compiègne, France, 2008
19. FILLATRE Lionel, NIKIFOROV Igor, CASAS HERNANDEZ Pedro, VATON Sandrine Optimal volume anomaly detection in network traffic flows. EUSIPCO'08 : 16th European Signal Processing Conference, august 25-29, Lausanne, Switzerland, 2008
20. SAOUTER Yannick, VATON Sandrine Fast estimation of the Frame Error Rate for error-correcting codes by the Generalized Error Impulse Method. Turbo Coding 2008 : 5th International Symposium on turbo codes, 1-5 september, Lausanne, Suisse, 2008, pp. 305-309
21. CASAS HERNANDEZ Pedro, VATON Sandrine An adaptive multi-temporal approach for robust routing. EuroFGI Workshop on IP QoS and Traffic Control, December 6-7, Lisbon, Portugal, 2007
22. DAHMOUNI Hamza, VATON Sandrine, ROSSÉ David A Markovian signature-based approach to IP traffic classification. MineNet 2007 : ACM Sigmetrics Workshop on Mining Network Data, June 12, San Diego, CA , 2007, pp. 29-34
23. SALEM Osman, VATON Sandrine, GRAVEY Annie An efficient online anomalies detection mechanism for high-speed networks. MonAM 2007 : Second IEEE Workshop on Monitoring, Attack Detection and Mitigation, November 5-6 , Toulouse, France, 2007, pp. 1-6
24. SALEM Osman, VATON Sandrine, GRAVEY Annie A novel approach for anomaly detection over high-speed networks. EC2ND : European Conference on Computer Network Defense, October 4-5, Heraklion, Greece, 2007
25. BERMOLEN Paola, VATON Sandrine, JUVA Ilmari Search for optimality in traffic matrix estimation : a rational approach by Cramer-Rao lower bounds. NGI'06 : 2nd Conference on Next Generation Internet Design and Engineering, 3-5 april ,Valencia, Spain, 2006, pp. 224-231
26. JUVA Ilmari, VATON Sandrine, VIRTAMO Jorma Quick traffic matrix estimation based on link count covariances. ICC 2006 : IEEE International Conference on Communications, June 11-15, Istanbul, 2006
27. DAHMOUNI Hamza, ROSSÉ David, MORIN B., VATON Sandrine Analytical Model for Performance Evaluation of GPRS/EDGE Multiservice Networks. MWCN 2005 : 7th IFIP International Conference on Mobile and Wireless Communications Networks, Marrakech, Morocco (19-21 septembre 2005), 2005
28. DAHMOUNI Hamza, MORIN Bertrand, VATON Sandrine Performance Modelling of GSM/GPRS Cells with Different Radio Ressource Allocation Strategies. WCNC (IEEE Wireless Communications and Networking Conference), 13-17 March 2005, New Orleans, USA, 2005

29. DAHMOUNI Hamza, ROSSÉ David, MORIN Bertrand, VATON Sandrine Impact of Data Traffic Composition on GPRS Performances. 19th International Teletraffic Congress, august 29 - september 2 2005, Beijing, China
30. JUVA Ilmari, VATON Sandrine, VIRTAMO Jorma Quick Traffic Matrix Estimation Based on Link Counts Covariances. 2nd EuroNGI Workshop On Traffic Engineering, Protection and Restoration for Next Generation Internet, 21-22 april , Rome, 2005
31. CHONAVEL Thierry, VATON Sandrine Least square and Bayesian deconvolution in the presence of side effects and correlated noise. ISIVC'04 : 2nd International Symposium on Image/Video Communications over fixed and mobile networks , July 7-9, Brest, France, 2004, pp. 355-358
32. MARAKOV Dmitry, VATON Sandrine Experience in Traffic Matrix Estimation : comparison and benchmarking of existing techniques. 2nd International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC'04), Brest, 7-9 july 2004, 2004
33. VATON Sandrine, BEDO J.S Network Traffic Matrix : how can one learn the prior distributions from the link counts only ?. IEEE International Conference on Communications (ICC 2004), Paris, june 2004, 2004
34. VATON Sandrine, GRAVEY Annie, BEDO Jean-Sébastien Network tomography : an iterative bayesian analysis. ITC 2003 : 18th International Teletraffic Congress, 31 août - 05 septembre 2003, Berlin, Allemagne, 2003, pp. 261-270
35. BERROU Claude, VATON Sandrine Computing the minimum distance of linear codes by the error impulse method. IEEE International Symposium on Information Theory (ISIT'02), Lausanne, Suisse, 2002
36. BERROU Claude, VATON Sandrine, JEZEQUEL Michel, DOUILLARD Catherine Computing the minimum distance of linear codes by the error impulse method.. IEEE GLOBECOM 2002, Taipei, Taïwan, 2002
37. VATON Sandrine, GRAVEY Annie Iterative bayesian estimation of network traffic matrices in the case of bursty flows. ACM/SIGCOMM Internet Measurement Workshop (IMW'02), Marseille, 6-8 novembre, France, 2002
38. VATON Sandrine, CHONAVEL Thierry, SAOUDI Samir Approximate and exact ML detectors for CDMA and MIMO systems : a tree detection approach. Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT'02), Rennes, France, 2002
39. VATON Sandrine, CHONAVEL Thierry, SAOUDI Samir Iterative multi-user detection and decoding for turbo-coded DS-CDMA systems. XIth European Signal Processing Conference (Eusipco 2002), 03-06 september 2002, Toulouse, France, 2002
40. SALAMATIAN Kave, VATON Sandrine Hidden Markov modelling for network communication channels. ACM Sigmetrics 2001 Conference, Cambridge, Massachussets, USA, 2001
41. VATON Sandrine Near completely decomposable Markov models of traffic. Eighth IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks, Bradford, 2000
42. VATON Sandrine, CHONAVEL Thierry Estimating the offset parameters of a mixture in the Fourier domain. ICASSP'99 (IEEE international conference on acoustics, speech and signal processing).Phoenix, March 15-19, 1999, pp. 1553-1556
43. VATON Sandrine A new test for stationarity and its application to teletraffic data. ICASSP'98 (IEEE international conference on acoustics, speech and signal processing), Seattle, USA, 1998
44. VATON Sandrine, MOULINES Eric A locally stationnary semi-markovian representation of LAN traffic data. IFIP TC6/WG6.2 Fourth International Conference on Broadband Communications (BC '98), Stuttgart, 1998
45. VATON Sandrine, MOULINES E., KOREZLIOGLU H., KOFMAN Daniel Statistical identification of LAN traffic data. Fifth IFIP Workshop on Performance Modelling and Evaluation in ATM Networks, July 21-23, Ilkley, UK, 1997

3.2 Conférences Nationales avec Comité de Lecture

1. FILLATRE Lionel, NIKIFOROV Igor, VATON Sandrine Détection-localisation séquentielle d'anomalies volumiques dans un réseau. GRETSI 2007 : 21ème Collloque sur le Traitement du Signal et des Images, 11-14 septembre, Troyes, France, 2007
2. VATON Sandrine, CHONAVEL Thierry Estimation d'un mélange de lois décalées dans le domaine de Fourier. XXXèmes Journées de la statistique, ENSAI, Rennes, France, 1998

3.3 Conférences Invitées

1. VATON Sandrine Statistical Inference of the Origin Destination Traffic Matrix : existing techniques and our Bayesian method. IX Congreso Latino Americano de Probabilidad y Estadistica Matematica, Punta del Este, Uruguay, mars 2004
2. VATON Sandrine Estimation de matrices de trafic : bilan des techniques existantes. Journées MAS (Modélisation Aléatoire et Statistique) de la SMAI (Société de Mathématiques Appliquées & Industrielles), septembre 2004, Nancy, 2004
3. VATON Sandrine 'Fractal' versus 'Markov' models of traffic. 8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks, July 17-19, Bradford, UK, 2000

4 Thèse de doctorat

– VATON Sandrine, Modélisation statistique de trafic dans les réseaux, Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications (TELECOM Paris), Spécialité : Signal et Images, Thèse soutenue le 11 décembre 1998.

5 Rapports Internes de Recherche

1. VATON Sandrine, CHONAVEL Thierry, Estimating the offset parameters of a mixture in the Fourier domain, Rapport Interne de Recherche, Télécom Bretagne, RR-200001-SC, Version longue d'un article ICASSP 2009.
2. VATON Sandrine, Some alternatives to long-range dependent models of traffic, Rapport Interne de Recheche, Télécom Bretagne, RR-2000002-ELEC.

6 Rapports de stage

1. VATON Sandrine, Prix des Calls et Dynamique du Sous-Jacent, Rapport de stage de DEA "Probabilités et Applications" (Paris 6), Société Générale MARC/SGOP/R&D, Septembre 2005
2. VATON Sandrine, Rapport de Stage Ingénieur Télécom Paris, Société Générale, Direction des Marchés de Capitaux, Septembre 2004
3. VATON Sandrine, Telefunken System Technik, Ulm (Allemagne), Unité de R&D Radars, Rapport de Stage Recherche, Télécom Paris, Septembre 2002

7 Rapports de Contrats de Recherche

Nombreux rapports dans le cadre des contrats de recherche. Voir la Section "Contrats de Recherche" dans le Curriculum Vitae. Les rapports sont généralement disponibles sur le site Web public des contrats ; par exemple pour certains contrats en cours : ETICS `https://www.ict-etics.eu/`, DEMONS `http://fp7-demons.eu/`, VIPEER `http://recherche.telecom-bretagne.eu/vipeer/`.

INTRODUCTION

EVOLUTIONS RECENTES DANS LE DOMAINE DES RESEAUX

J'ai démarré ma thèse de doctorat au milieu des années 1990 sur la modélisation statistique de trafic réseau. A cette époque le domaine a connu une évolution majeure. En effet des analyses de traces de réseau local [32] [33] initiées au Lawrence Berkeley Laboratory ont montré que les hypothèses classiques de modélisation markovienne étaient erronées. Ces hypothèses classiques (arrivées poissonniennes, durées de services exponentielles) dataient des premiers travaux d'Erlang au début du XXème siècle et justifiaient le dimensionnement des systèmes par des modèles de filtes d'attente markoviennes [1].

Les travaux du milieu des années 1990 [32] [33] ont montré que les hypothèses classiques étaient erronées. Au contraire ils ont mis en évidence certaines propriétés inédites comme la présence de rafales à plusieurs échelles de temps, d'auto-similarité, ou encore une décroissance très lente de la fonction d'auto-corrélation du trafic. Ces travaux ont eu un retentissement important car on en a conclu que les modèles classiques de files d'attente étaient extrêmement optimistes et que les performances réelles (délais, pertes, etc...) étaient dramatiquement plus mauvaises que ce que prédisait la théorie. Depuis on a relativement bien compris l'origine du phénomène de dépendance longue qu'on explique comme le résultat de l'agrégation d'un grand nombre de flux correspondant au téléchargement de fichiers dont la taille suit une loi à queue lourde [43].

Depuis l'époque où j'ai démarré ma thèse le contexte de la recherche dans le domaine des réseaux de télécommunications et en particulier de l'Internet a considérablement évolué. Tout d'abord du point de vue des services, les années 2000 ont vu se mettre en place la convergence sur l'Internet de trois types de service, les données, le téléphone et la télévision, et l'apparition de nouvelles applications (en particulier la vidéo) extrêmement gourmandes en bande passante.

Sur le plan de l'accès, le paysage a également changé. Le développement de l'accès fixe à l'Internet à très haut débit grâce au déploiement des technologies optiques jusqu'à l'usager terminal (FTTH) est de nature à stimuler l'utilisation de services à forte bande passante. En France la loi de modernisation de l'économie de 2008 a donné un cadre juridique au déploiement par les opérateurs de la fibre jusqu'à l'usager terminal. Cette évolution dans l'accès résidentiel induit une évolution des usages vers des applications demandeuses en bande passante et en Qualité de Service (QoS). Par ailleurs les dernières années ont vu l'avènement de l'accès mobile à Internet avec les technologies GPRS/Edge, UMTS, et actuellement LTE et une multiplication de l'offre de terminaux d'accès mobiles. Les services deviennent ubiquitaires et les internautes vont avoir la possibilité de rester connectés tout en faisant de la mobilité et en basculant entre plusieurs types de réseaux d'accès (WiFi domestique, hotspot, 3G).

L'émergence de nouvelles applications (vidéo, voix, jeux ...) a considérablement modifié la nature du trafic Internet. Ainsi le trafic pair-à-pair (P2P) était au milieu des années 2000 le trafic majoritaire, supplantant le trafic généré par les applications plus anciennes de navigation Web ou de messagerie électronique. Puis le trafic P2P a progressivement perdu de son importance pour être supplanté par le trafic de vidéo en streaming (vidéo à la demande, télévision en direct ou en rattrapage, video en streaming). Le streaming représente à la fin des années 2000 la majorité du trafic Internet. Du fait de l'émergence des nouvelles applications la nature du trafic est donc en permanente mutation.

La distribution spatiale du trafic est également en permanente évolution. Avec le développement du P2P les ordinateurs sont devenus à la fois clients et serveurs contrairement au modèle client/serveur où les données étaient localisées sur des serveurs et accédées via le réseau par les machines clientes. Les CDN (*Content Delivery Networks*) participent à la distribution d'une grande partie du contenu multimédia (Youtube, etc...) et ont un impact majeur sur la distribution spatiale de la demande sur les réseaux et sur

1. Lire par exemple l'article de François Baccelli Internet : modéliser le trafic pour mieux le gérer, Interstices.info, décembre 2005

les volumes de trafic sur les liens d'interconnexion. Le développement de l'informatique dans les nuages (*cloud computing*) qui est actuellement en pleine expansion est également en train de modifier le paysage. Dans ce modèle les données et les applications de l'entreprise ou de l'utilisateur sont externalisées chez Google, Amazon, MicroSoft ou dans des *clouds* privés et accédées via le réseau. Elles sont en pratique hébergées dans quelques immenses centres de données (*data centers*) et accessibles depuis n'importe quel terminal connecté à l'Internet. La montée en puissance des services de *cloud computing* présente un enjeu majeur dans le domaine de la gestion énergétique de ces centres de données, de la gestion du trafic, et de la sécurisation des données.

En même temps que, stimulés en particulier par le développement des technologies FTTH et de l'Internet mobile, de nouvelles applications ont émergé et sont montées en puissance, induisant une croissance très rapide en volume et en hétérogénéité du trafic Internet, les modèles de stockage et de distribution des contenus ont évolué avec l'émergence du P2P, des CDN et du cloud computing. Du fait de l'augmentation des débits, pour éviter la congestion et offrir une qualité de service correcte aux internautes, les opérateurs de télécommunications (fournisseurs d'accès, opérateurs de coeurs de réseau) sont contraints d'investir dans leurs infrastructures. Mais la seule augmentation des capacités ne suffit plus à maintenir la qualité de service. Les opérateurs sont contraints d'optimiser la gestion du trafic dans leurs réseaux pour mieux le contrôler. Ils choisissent aussi de contrôler eux-mêmes les sources de trafic, par exemple en déployant eux-mêmes des CDN dont les noeuds sont installés au plus près de leurs abonnés (TelCo CDNs). Le projet ANR VIPEER [3] conduit par Télécom Bretagne et dans lequel je suis actuellement impliquée étudie ainsi la possibilité d'un CDN opéré par le fournisseur d'accès.

Les nouvelles applications (vidéo, jeux, etc...) ont des besoins différents en QoS que l'Internet de base qui fonctionne selon un mode Best Effort ne peut pas garantir. La bande passante est devenue une ressource relativement rare même dans les coeurs de réseau. Il est donc nécessaire d'adapter le fonctionnement du réseau au trafic et de différencier les services. Toutefois les opérateurs ne doivent pas enfreindre le principe de neutralité du réseau (*Net neutrality*) qui garantit un accès à l'Internet pour tous sans discrimination des flux et sans entorse au principe de libre concurrence. La neutralité du réseau fait l'objet actuellement de vifs débats politiques. En France l'ARCEP (Autorité de régulation des communications électroniques et des postes) a engagé une réflexion avec consultation publique sur le sujet. De leur côté Vivendi, Alcatel-Lucent et Deutsche Telekom ont fait en juillet 2011 des propositions en faveur de la différenciation de services à la commissaire européenne en charge du numérique dans le cadre d'une consultation sur les mesures à mettre en oeuvre pour atteindre les objectifs de l'agenda numérique en 2020.

Une autre évolution majeure des usages de l'Internet a vu le jour dans le courant des années 2000 avec l'émergence des réseaux sociaux. Au-delà des seuls aspects techniques l'émergence des réseaux sociaux a été avant tout une révolution culturelle. Les réseaux sociaux ont démontré très récemment leur extrême importance sociale puisqu'ils ont joué le rôle de catalyseurs de mouvements de masse lors du Printemps Arabe. Les Etats ont bien compris l'importance politique de l'Internet qui est vu comme une menace par certains états totalitaires comme le montrent les blocages récents de l'Internet en Lybie ou en Egypte (voir les données du Google Transparency Report [27]).

Les dix dernières années ont vu une multiplication des attaques contre les infrastructures numériques, certaines d'entre elles étant très médiatisées. La multiplication des terminaux et des services a introduit autant de sources nouvelles de vulnérabilités. Les attaques sont de plus en plus sévères et de plus en plus sophistiquées. Les attaques contre les infrastructures numériques (sites gouvernementaux, sites de e-commerce, systèmes d'information de grandes entreprises, etc...) sont vues comme une arme extrêmement efficace et médiatisée à l'ère de la société de l'information. Les motivations des cyberattaques sont variées ; elles peuvent aller de l'extorsion de fonds, au vol d'informations sensibles (personnelles, industrielles) ou à des motivations politiques.

LES ENJEUX ACTUELS

Internet repose sur des principes fondateurs très simples ayant abouti à la mise en place très rapide d'un réseau à l'échelle mondiale. On est passé en une quarantaine d'années des balbutiements du routage et de l'interconnexion de réseaux à une toile à l'échelle planétaire, supportant une grande variété d'applications et supplantant en grande partie les infrastructures traditionnelles de communication. Le modèle actuel est

centré sur une couche réseau capable de sélectionner dynamiquement un chemin d'une source de paquets à sa destination sans garantie sur la QoS ou sur la forme du trafic. Cependant l'Internet actuel présente un certain nombre de limites bien identifiées par la communauté scientifique dans le domaine de la sécurité, de la QoS, de l'ubiquité, du passage à l'échelle, de la vulnérabilité de certains services cruciaux (DNS).

Dans le cadre des travaux sur les réseaux du futur une part importante de la communauté scientifique réfléchit aux évolutions graduelles (ou incrémentales) ou à la refondation de l'Internet. Deux approches s'opposent : l'approche incrémentale et l'approche de refondation ou *clean slate*. L'approche *clean slate* (du nom d'un programme de recherche initié à l'université de Stanford) envisage une refondation de l'Internet avec la création de nouveaux réseaux et protocoles capables de supporter la croissance continue des nouveaux usages. L'approche incrémentale est l'approche évolutive suivie actuellement ; elle maintient le fonctionnement actuel de l'Internet en proposant des solutions ponctuelles ce qui conduit à une complexification du fonctionnement du réseau. Une troisième approche propose de construire des réseaux overlays qui profitent des infrastructures existantes pour construire de nouveaux réseaux et protocoles.

L'Internet actuel se trouve confronté à un certain nombre d'enjeux. Un premier challenge est lié à l'explosion du nombre d'objets à intégrer dans l'Internet comme les capteurs et autres objets communicants qui seraient de l'ordre du milliard, et à une mobilité généralisée. Il est connu depuis longtemps que l'espace d'adressage offert par IPv4 est largement insuffisant ; ceci est devenu récemment une réalité, les dernières plages d'adresses ayant été attribuées en 2011 par l'IANA. Il devient donc extrêmement urgent de migrer vers IPv6 ; le déploiement d'IPv6 sur Internet est compliqué en raison de l'incompatibilité des adresses IPv4 et IPv6 et de la nécessité de mettre en place des solutions de type tunnels pour gérer le cas des routeurs ne supportant pas encore IPv6.

Les exigences en QoS de certaines applications comme la voix, la vidéo, la télé-médecine ou les jeux est un autre challenge pour le monde de l'Internet. Les paramètres de Qualité de Service qu'il convient de contrôler sont la bande passante, la disponibilité, la latence ... Il est nécessaire d'optimiser le réseau pour contrôler ces paramètres. En particulier la latence est devenu un paramètre critique pour des applications qui manipulent des masses de données considérables et en temps réel comme le Cloud Computing ou les applications financières [24] [34].

D'un côté il est nécessaire de différencier les services mais de l'autre on doit continuer à garantir l'accès à l'Internet pour tous sans discrimination. Comme son nom l'indique l'Internet est formé d'une interconnexion de dizaines de milliers de systèmes autonomes. La qualité d'expérience perçue par de les utilisateurs est liée aux performances offertes de bout-en-bout par cette interconnexion de réseaux. Pour offrir des services avec QoS toute la chaîne de distribution du trafic depuis sa source jusqu'à sa destination doit être prise en compte. Il est nécessaire de concevoir des solutions techniques permettant l'interconnexion de réseaux avec garantie de QoS, les solutions proposées jusqu'à présent pour la QoS en inter-domaine ayant démontré leurs limitations. De plus, dans la mesure où la distribution de contenus avec garanties de QoS est une source de revenus considérable, il est nécessaire de concevoir des modèles économiques permettant une redistribution des revenus incitant tous les acteurs de la chaîne à collaborer. Dans cette vision une partie de l'Internet devient une ressource qu'il va être possible de louer en composant des ressources offertes par les différents domaines pour obtenir des tuyaux de bout-en-bout avec qualité garantie. L'élaboration de solutions techniques et économiques pour les services garantis en inter-domaine est le sujet du projet intégré européen ETICS, Economics and Technologies for InterCarrier Services [22], auquel je participe.

L'apparition massive de nouveaux dispositifs et la densité croissante des utilisateurs et des données échangées exacerbent les problèmes de passage à l'échelle à tous les niveaux : gestion de la mobilité et des ressources, sécurité, qualité de service, plan de contrôle intra et inter-domaine, networking sensible au contexte et au contenu, etc ... En particulier il est devenu nécessaire d'automatiser la gestion du réseau et de concevoir des mécanismes permettant aux services et au réseau d'adapter leur fonctionnement à l'état du réseau et à la nature du trafic. Pour cela il faut que les services et le réseau soient conscients de leurs propres conditons de fonctionnement et de leur environnement (*network awareness*, *service awareness*, *self awareness*). Il est nécessaire d'intégrer plus d'intelligence dans les équipements de réseau, par exemple dans les routeurs avec le développement de routeurs cognitifs ou sémantiques. A la différence du routage classique qui consiste à router des messages d'un site source à une destination connue, le routage sémantique achemine les requêtes utilisateurs vers des sites pouvant répondre «au mieux» à leur besoin d'information en se basant uniquement sur le contexte de la requête. Les routeurs cognitifs basent les décisions de routage sur l'adresse de destination des paquets mais aussi sur la connaissance qu'ils ont de

la qualité des liens sur les chemins en direction de la destination.

Les montées en débit ainsi que la complexification liées à l'émergence des nouvelles applications nécessitent de concevoir des approches logicielles et matérielles qui soient à la fois flexibles et performantes. Ces approches doivent permettre des traitements sophistiqués et réactifs et être capables de supporter la montée en débit du trafic. On peut à la fois s'appuyer sur les avancées récentes dans le domaine des architectures multi-coeur et sur les possibilités offertes par l'accélération matérielle (FPGA, ASIC) pour atteindre ces objectifs. Dans le cadre du projet intégré européen DEMONS [21] nous développons Blockmon, un environnement permettant d'intégrer simplement des fonctions de surveillance du trafic à des débits allant jusqu'à 12 millions de paquets par seconde. Pour cela nous utilisons à la fois du parallélisme au niveau logiciel et des cartes spécialisées dans les applications réseau (NetFPGA, COMBOv2).

Les processeurs de réseau sont également une technologie intéressante. Cette technologie offre une solution souple et configurable pour s'adapter rapidement aux nouveaux protocoles. Ces processeurs spécialisés pour les systèmes de télécommunications possèdent des interfaces de types télécom, des processeurs embarqués orientés sur les protocoles (IP, ATM, Ethernet) et une très grande rapidité.

IMPORTANCE DE LA METROLOGIE

La métrologie des réseaux est la discipline qui consiste à mesurer diverses quantités relatives au trafic et à l'état du réseau, à analyser le résultat de ces mesures et éventuellement à en déduire une réaction. Une part importante des travaux que j'ai effectués depuis ma thèse de doctorat concerne la métrologie des réseaux, c'est pourquoi j'ai articulé ce manuscrit d'Habilitation à Diriger les Recherches autour de certaines contributions dans ce domaine.

Tout d'abord on a besoin de caractériser le trafic pour en connaître les propriétés et ainsi mieux en comprendre l'impact sur le réseau pour adapter au mieux l'ingénierie du réseau aux propriétés du trafic. Un exemple typique concerne les travaux sur les propriétés d'auto-similarité du trafic [32] [33] et leur impact sur les performances attendues (délais, taux de perte, ...) ainsi que la découverte de l'origine de ces propriétés d'auto-similarité dans la distribution des tailles de fichiers [43].

De la même façon il est nécessaire de caractériser les différents services pour évaluer la pression que ceux-ci font peser sur la bande passante des opérateurs. Dans le cadre de sa thèse CIFRE réalisée chez France Télécom, Hamza Dahmouni a travaillé sur la modélisation des services de données sur les réseaux GPRS et sur un modèle de performances voix+données pour ce réseau [18][17]. La mesure intéresse l'opérationnel puisque l'introduction de nouveaux services a un impact sur les services plus anciens et puisque les services ont des besoins différents en QoS. Elle intéresse également le marketing afin de connaître la popularité des services et d'identifier de nouvelles tendances dans les usages de l'Internet.

On peut proposer une taxonomie de la métrologie des réseaux selon différents critères : métrologie active versus métrologie passive, analyses hors ligne ou en temps réel, métrologie appliquée à la mesure de QoS ou à la découverte de problèmes de sécurité, mesure du trafic ou mesure de QoS, mesures au niveau paquet, flot, applicatif, mesure en un seul point ou distribuée, etc ... La caractérisation du trafic peut se faire à différents niveaux : à un grain très fin avec une modélisation au niveau paquet ou au contraire à très gros grain comme dans l'estimation de la matrice de trafic Origine/Destination à partir de mesures SNMP. Les mesures peuvent être faites au niveau des différents réseaux d'accès (mobile, fixe ADSL, FTTH), au niveau des réseaux de collecte ou des coeurs de réseau. La mesure peut se limiter à un seul système autonome ou au contraire tenter de caractériser le réseau au niveau inter-domaine. On peut s'intéresser également à la caractérisation du trafic de différents réseaux overlays comme les réseaux privés virtuels (VPN) ou les réseaux spontanés P2P.

La métrologie vise à caractériser la demande de trafic mais également la topologie ou l'état du réseau. Un certain nombre d'études se sont intéressées à la caractérisation de la topologie de l'Internet et à la production d'un graphe de l'Internet. La Qualité de Service (latence, taux de perte, bande passante disponible, réordonnancement ou duplication de paquets, etc...) doit pouvoir être mesurée, par exemple pour vérifier des accords de service et mettre en place des contre-mesures en cas de violation. Le groupe de travail IPPM (IP Performance Metrics) [29] à l'IETF travaille à la standardisation des méthodologies de mesure de la performance sur l'Internet.

Mais la qualité des communications est avant tout la qualité perçue par les usagers des différents services multimedia (QoE, *Quality of Experience*). La QoE dépend de beaucoup de facteurs, au niveau

du réseau et au niveau multimédia (codecs). La mesure de la qualité d'expérience pour les applications multimédia est un domaine très actif.

Dans les approches cognitives ou *-*aware* (trafic awareness, network awareness) les équipements de réseau et les services doivent être conscients de leur état et de l'état de leur environnement pour adapter leur fonctionnement au contexte. Cette prise de conscience passe par différentes mesures du trafic et du réseau. Dans ce cas l'utilisation de services de monitoring vient en support des fonctionnalités d'auto-configuration des systèmes.

Une autre application importante de la métrologie concerne la sécurité des réseaux. Les politiques de sécurité ne sont pas parfaites et ne permettent pas à elles seules de garantir la sécurité du réseau. Les systèmes de détection d'intrusion de niveau réseau (NIDS, *Network Intrusion Detection Systems*) écoutent le trafic de réseau et y détectent des signatures spécifiques ou des comportements statistiquement suspects.

La mesure ne sert pas uniquement à la gestion et à la surveillance du réseau. Les mesures deviennent une valeur marchande car les utilisateurs révèlent des informations personnelles via leur utilisation des réseaux, et ces informations peuvent être exploitées à des fins commerciales. A titre d'exemple les données de mobilité des usagers dans les réseaux cellulaires (handovers, etc...) peuvent être utilisées par l'opérateur pour construire des modèles de déplacement de population. Ces modèles intéressent les collectivités locales dans le cadre de la planification des réseaux de transport urbain. Le comportement de téléchargement des usagers est analysé pour pousser des contenus multimédias ou de la publicité de manière anticipée vers ces usagers. La diffusion d'informations sur les réseaux sociaux (buzz, e-reputation...) peut être mesurée et modélisée pour servir à des fins de e-marketing [30]. L'exploitation de données de nature personnelle obtenues par l'analyse des communications sur le réseau pose certainement des problèmes éthiques et juridiques.

CONTENU ET STRUCTURE DU DOCUMENT

Dans ce manuscrit d'Habilitation à Diriger les Recherches je présente quelques travaux de recherche développés sur les dernières années dans le domaine de la métrologie des réseaux. J'ai choisi, pour des raisons de concision et de cohérence du document, de ne présenter que quelques-uns des axes sur lesquels j'ai travaillé depuis l'obtention de la thèse de doctorat. Les critères qui ont guidé ce choix sont doubles. D'une part j'ai choisi de présenter des travaux récents, obtenus en collaboration avec des doctorants et post-doctorants et dans le cadre de projets de recherche au niveau national ou européen. D'autre part je présente des travaux relatifs à des axes de recherche dont j'assurais la direction, sur lesquels plusieurs articles de bon niveau ont été publiés, et qui sont relatifs à un axe de recherche maintenu sur plusieurs années.

J'ai fait le choix de ne pas présenter des travaux intéressants mais plus anciens ou à la marge de mes activités actuelles. Par exemple mes travaux dans le domaine des communications numériques ne sont pas mentionnés dans ce document : ils sont relatifs au codage de canal [25][26], à l'estimation performante des taux d'erreur des turbo-codes [6] [7] [38], aux techniques de décodage en CDMA [42]. Il s'agit d'une activité de recherche ancienne qui n'a pas été poursuivie. De la même façon les travaux sur la modélisation de trafic dans les réseaux d'accès mobile et sur les modèles de performance pour des réseaux mobiles voix+données (GPRS) [18][17] ne sont pas présentés. Ces travaux ont été réalisés dans le cadre de la thèse CIFRE de Hamza Dahmouni chez Orange Labs et étaient liés au contexte particulier du contrat d'encadrement doctoral. Par ailleurs de nombreux travaux dans le domaine de la modélisation de trafic ou de réseaux ne sont pas mis en avant, toujours en raison de la nécessité de ne dégager que quelques axes principaux [35] [41] [23] ...

J'ai stucturé ce manuscrit autour de trois axes : (1) la matrice de trafic et le routage robuste (2) la classification de trafic (3) la détection d'anomalies dans le trafic. Chacun de ces trois chapitres fait un état de l'art du domaine et présente nos contributions en s'appuyant sur quelques publications principales inclues dans le manuscrit.

Le premier axe concerne la matrice de trafic [40] [31] [5] [13] [12] [11], le routage robuste et l'ingénierie de trafic en présence d'incertitude sur la demande [10] [9] [2]. Il a donné lieu aux thèses de doctorat de Dmitry Marakov et Pedro Casas, au séjour post-doctoral de Lionel Fillatre, et à des collaborations dans le cadre du réseau d'excellence EuroNGI et d'un projet ECOS-Sud avec l'Uruguay. La dernière de ces publications [2] est un travail réalisé avecma doctorante Isabel Amigo dans le cadre du projet européen

ETICS : *Economics and Technologies for InterCarrier Services* [22].

Le deuxième axe sur la classification de trafic a été initié à la fin de la thèse de Hamza Dahmouni [19]. Il se poursuit par un travail réalisé dans la thèse de Tristan Groléat sur des aspects relatifs à l'accélération matérielle et des performances d'une implémentation de la classification de trafic basée sur une méthode SVM (*Support Vector Machines*) [28]. Sur la classification de trafic et plus généralement l'analyse de mesures de trafic j'organise annuellement le workshop TRAC : *International Workshop on Traffic Analysis and Classification* [14] [15] [16] (avec Christian Callegari, université de Pise).

Enfin le troisième axe concerne des travaux autour de la détection d'anomalies dans le trafic et plus généralement, de la métrologie appliquée à la sécurité. Après des premiers travaux avec Christian Callegari de l'université de Pise [8] cet axe de recherche a été poursuivi dans le cadre du projet ANR OSCAR : *Overlay Networks Security : Characterization, Analysis and Recovery* [4] et du post-doctorat de Osman Salem [36] [37]. Nous collaborons actuellement au projet intégré européen DEMONS : *DEcentralized, cooperative and privacy-preserving MONitoring for trustworthinesS* [21] dans le cadre duquel Tristan Groléat réalise sa thèse de doctorat. Nous participons en particulier au développement de Blockmon, un système modulaire pour la surveillance de trafic flexible et à haute performance [20].

Enfin dans la dernière partie du document je dégage des perspectives de recherche. Je fais tout d'abord une analyse des limites que j'identifie dans la recherche académique en métrologie des réseaux. J'analyse pour quelles raisons ce domaine de recherche très actif a conduit, selon moi, à peu de transfert technologique. De cette analyse critique je dégage des points d'action potentiels. Puis j'expose plus en détails un projet de mise en place d'une plateforme pour le monitoring de trafic à haut débit afin de travailler sur le passage à l'échelle en débit des algorithmes d'analyse du trafic. Je présente ensuite des travaux de recherches en cours dans le cadre de la thèse d'Isabel Amigo [1] et du projet intégré européen ETICS, *Economics and Technologies for InterCarrier Services* [22] [39] sur l'analyse économique de l'interconnexion de réseaux. J'expose également une idée de projet européen dont l'objectif est de stimuler l'enseignement dans le secondaire et dans le supérieur de la science des réseaux en mettant en place une plateforme offrant un certain nombre de supports de cours et de services adaptés. Je présente aussi une idée de projet d'observatoire de l'Internet à l'échelle nationale. Puis je conclue sur ma vision des évolutions futures de la métrologie des réseaux.

Références

[1] I. Amigo, P. Belzarena, F. Larroca, and S. Vaton. Network bandwidth allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations. In *ICQT'11 : 7th International ICQT Workshop on Advanced Internet Charging and QoS Technology*, Heidelberg, Germany, 2011.

[2] I. Amigo, S. Vaton, T. Chonavel, and F. Larroca. Maximum delay computation for interdomain path selection. *International Journal of Network Management*, 22(2) :162–179, march/april 2012.

[3] ANR Programme VERSO (ANR-09-VERS-014). VIPEER : Video Traffic Engineering in an Intra-Domain Context using P2P Paradigms. `http://recherche.telecom-bretagne.eu/vipeer/`.

[4] ANR RNRT (projet labellisé fin 2005). OSCAR : Overlay Security, Characterization, Analysis and Recovery.

[5] P. Bermolen, S. Vaton, and I. Juva. Search for optimality in traffic matrix estimation : a rational approach by Cramer-Rao lower bounds. In *NGI'06 : 2nd Conference on Next Generation Internet Design and Engineering*, 2006.

[6] C. Berrou and S. Vaton. Computing the minimum distance of linear codes by the error impulse method. In *IEEE International Symposium on Information Theory*, juillet 2002.

[7] C. Berrou, S. Vaton, M. Jezequel, and C. Douillard. Computing the minimum distance of linear codes by the error impulse method. In *IEEE GLOBECOM 2002*, novembre 2002.

[8] C. Callegari, S. Vaton, and M. Pagano. A new statistical method for detecting network anomalies in tcp traffic. *European Transactions on Telecommunications*, 21(7) :575–588, 2010.

[9] P. Casas, L. Fillatre, S. Vaton, and I. Nikiforov. Reactive robust routing : Anomaly localization and routing reconfiguration for dynamic networks. *Journal of Network and System Management*, 19(1) :58–83, 2011.

[10] P. Casas, F. Larroca, J.L. Rougier, and S. Vaton. Taming traffic dynamics : Analysis and improvements. *Computer Communications*, 35 :565–578, march 2012.

[11] P. Casas and S. Vaton. On the use of random neural networks for traffic matrix estimation in large-scale IP networks. In *Proceedings of the 1st Int. Workshop on Traffic Analysis and Classification*, IWCMC'10, 2010.

[12] P. Casas, S. Vaton, L. Fillatre, and T. Chonavel. Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks. In *ITC'21 : 21st conference on International Teletraffic Congress*, 2009.

[13] P. Casas, S. Vaton, L. Fillatre, and I. Nikiforov. Optimal volume anomaly detection and isolation in large-scale ip networks using coarse-grained measurements. *Computer Networks*, 54(11) :1750–1766, 2010.

[14] C.Callegari and S.Vaton (Chairs). TRAC'10 : 1st International Workshop on Traffic Analysis and Classification. `http://netgroup.iet.unipi.it/trac2010/Home_Page.html`.

[15] C.Callegari and S.Vaton (Chairs). TRAC'11 : 2nd International Workshop on Traffic Analysis and Classification. `http://netgroup.iet.unipi.it/trac2011/Home_Page.html`.

[16] C.Callegari and S.Vaton (Chairs). TRAC'12 : 3rd International Workshop on Traffic Analysis and Classification. `http://netgroup.iet.unipi.it/trac2012/Home_Page.html`.

[17] H. Dahmouni, B. Morin, and S. Vaton. Performance modelling of GSM/GPRS cells with different radio ressource allocation strategies. In *WCNC 2005 : IEEE Wireless Communications and Networking Conference*, 2005.

[18] H. Dahmouni, D. Rossé, B. Morin, and S. Vaton. Impact of data traffic composition on GPRS performances. In *19th International Teletraffic Congress*, 2005.

[19] H. Dahmouni, S. Vaton, and D. Rossé. A markovian signature-based approach to IP traffic classification. In *MineNet 2007 : ACM Sigmetrics Workshop on Mining Network Data*, pages 29 – 34, 2007.

[20] Di Pietro, A. and Huici, F. and Bonelli, N. and Kastovsky, P. and Trammell, B. and Groleat, T. and Vaton, S. and Nuccilli, F. Blockmon : a modular system for flexible, high-performance traffic monitoring and analysis. soumis à Usenix 2012.

[21] European Commission 7th Framework Program (FP7). DEMONS : DEcentralized MONitoring for trustworthinesS . `http://fp7-demons.eu/`.

[22] European Commission, Information Society Technologies, Seventh Framework Program. ETICS : Economics and Technologies for InterCarrier Services. `https://www.ict-etics.eu/`.

[23] L. Fillatre, D. Marakov, and S. Vaton. Forecasting seasonal traffic flows. In *EuroNGI Workshop on QoS and Traffic Control*, 2003.

[24] Gigaom. Structure 2010 : Latency Still a Problem For the Cloud. `http://gigaom.com/2010/06/24/structure-2010-latency-still-a-problem-for-the-cloud/`.

[25] A. Glavieux, P. Adde, G. Battail, E. Biglieri, M. Jezequel, A. Poli, S. Vaton, R. Pyndiah, A. Picart, C. Douillard, and C. Berrou. *Codage de canal - des bases théoriques aux turbocodes (sous la direction de Alain GLAVIEUX)*. Traité IC2 : Traitement du signal et de l'image. Hermès Science / Lavoisier, Paris, 2005.

[26] A. Glavieux, P. Adde, G. Battail, C. Douillard, A. Picart, R. Pyndiah, S. Vaton, A. Poli, M. Jezequel, and E. Biglieri. *Channel coding in communication networks : from theory to turbocodes. Edited by Alain Glavieux.* ISTE Ltd, London, 2007.

[27] Google Transparency Report. Google Transparency Report : Lybie. `http://www.google.com/transparencyreport/traffic/?r=LY&l=EVERYTHING&csd=1305365724070&ced=1313901000000`.

[28] Groleat, T. and Martinez, S. and Sbai, M.K. and Vaton, S. and Arzel, M. and Guelton, S. Practical Implementation of SVM-Based Traffic Classification : Dynamic Learning, Software Parallelization using GPU, and Hardware Acceleration on FPGA. Soumis à Sigmetrics 2012.

[29] IETF Internet Engineering Task Force. IP Performance Metrics (IPPM) Working Group. `http://datatracker.ietf.org/wg/ippm/charter/`.

[30] Institut Telecom. Chaire Reseaux Sociaux. `http://chaireresеaux.wp.institut-telecom.fr/`.

[31] I. Juva, S. Vaton, and J. Virtamo. Quick traffic matrix estimation based on link counts covariances. In *2nd EuroNGI Workshop On Traffic Engineering, Protection and Restoration for Next Generation Internet*, 2005.

[32] W.E. Leland, W. Willinger, M.S. Taqqu, and D.V. Wilson. On the selfsimilar nature of ethernet traffic. In *In Proc. of the ACM SIGCOMM'93*, 1993.

[33] V. Paxson and S. Floyd. Wide-area traffic : The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3 :226–244, 1995.

[34] Popsi : the future now. Fiber-optic transatlantic cable could save milliseconds, millions by speeding data to stock traders. `http://www.popsci.com/technology/article/2011-04/new-transatlantic-cable-will-speed-information-exchange-price`.

[35] K. Salamatian and S. Vaton. Hidden Markov modelling for network communication channels. In *ACM Sigmetrics Conference*, 2001.

[36] Osman Salem, Sandrine Vaton, and Annie Gravey. An Efficient Online Anomalies Detection Mechanism for High-Speed Networks. In *IEEE Workshop on Monitoring, Attack Detection and Migitation (MonAM 2007)*, November 2007.

[37] Osman Salem, Sandrine Vaton, and Annie Gravey. A scalable, efficient and informative approach for anomaly-based intrusion detection systems : theory and practice. *International journal of network management*, 20 :271–293, 2010.

[38] Y. Saouter and S. Vaton. Fast estimation of the frame error rate for error-correcting codes by the generalized error impulse method. In *Turbo Coding 2008 : 5th International Symposium on turbo codes*, pages 305 – 309, 2008.

[39] N. Le Sauze, A. Chiosi, R. Douville, H. Pouyllau, H. Lonsethagen, P. Fantini, C. Palas-ciano, A. Cimmino, M. A. Callejo Rodriguez, O. Dugeon, D. Kofman, X. Gadefait, P. Cuer, N. Ciulli, G. Carrozzo, A. Soppera, B. Briscoe, F. Bornstaedt, M. Andreou, G. Stamoulis, C. Courcoubetis, P. Reichl, I. Gojmerac, J. L. Rougier, S. Vaton, D. Barth, and A. Orda. ETICS : QoS-enabled interconnection for Future Internet services. In *Future Network and Mobile Summit*, 2010.

[40] S. Vaton, J.S Bedo, and A. Gravey. *Advanced methods for the estimation of the origin destination traffic matrix*, volume XVI, chapter Performance Evaluation and Planning Methods for the Next Generation Internet (25 th du GERARD). A. Girard, B. Sanso, F. Vazquez-Abad, 2005.

[41] S. Vaton and T. Chonavel. Estimating the offset parameters of a mixture in the Fourier domain. In *ICASSP : IEEE international conference on acoustics, speech and signal processing*, 1999.

[42] S. Vaton, T. Chonavel, and S. Saoudi. Iterative Multi-user Detection and Decoding for Turbo-coded DS-CDMA Systems. In *Eusipco'2002 : 11th European Signal Processing Conference*, Toulouse, Sep. 2002.

[43] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson. Self-similarity through high-variability : Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1) :71–86, 1997.

MATRICE DE TRAFIC

CONTEXTE, ETAT DE L'ART ET CONTRIBUTIONS

Positionnement du problème de la matrice de trafic

La matrice de trafic est une représentation des volumes de trafic échangés dans un réseau entre chaque noeud source et chaque noeud destination de trafic. La matrice de trafic est une information utile par exemple pour la planification des capacités à déployer dans le réseau, pour l'optimisation de l'équilibrage de charge dans le réseau, pour la vérification des accords de peering qui contractualisent les échanges de trafic entre les Systèmes Autonomes (*Autonomous Systems*, AS). Il est utile pour les opérateurs de réseau d'avoir une estimation de la demande de trafic Origine/Destination (OD) et d'en surveiller les variations, qu'il s'agisse des variations régulières sur un rythme jour/nuit par exemple ou de variations brutales (pannes, reconfiguration de routage).

Une solution actuellement déployée pour mesurer la demande de trafic OD se base sur des outils de mesure de niveau flot comme NetFlow de CISCO [12] ou des solutions équivalentes d'autres vendeurs (CFlowd pour Alcatel-Lucent, JFlow ou CFlowd pour Juniper). Un flot est défini comme une succession mono-directionnelle de paquets ayant un certain nombre de champs d'entête en commun ; la définition d'un flot selon CISCO se base sur un 7-uplet (IP Source, IP Destination, Port Source pour UDP ou TCP et 0 sinon, Port Destination pour UDP ou TCP ou Type et Code pour ICMP ou 0 sinon, protocole IP, interface d'entrée, Type de Service). Les rapports NetFlow contiennent des informations relatives au flot, en particulier des instants de début et fin de flot, le volume de trafic dans le flot (en octets et en paquet), les adresses IP source et destination.

Il est possible à partir des rapports de flots exportés par les équipements de réseau de reconstruire une estimation de la demande de trafic OD sur un AS. Cependant à l'époque de nos travaux cette solution basée NetFlow présentait un certain nombre de limitations. D'une part il était nécessaire de déployer NetFlow sur l'ensemble des routeurs ; cette mise à jour des routeurs a longtemps été un facteur de coût pour les opérateurs. D'autre part maintenir des données NetFlow peut être extrêmement lourd pour les routeurs. La charge supportée pour analyser chacun des paquets de façon à maintenir les données NetFlow peut être telle qu'on atteint les capacités de traitement du routeur, au détriment du routage. Pour remédier à ce problème, CISCO a développé une version échantillonnée de NetFlow (*sampled NetFlow*). Le principe est de ne plus analyser chaque paquet mais seulement un paquet tous les N (N ayant une valeur fixe ou aléatoire). Ceci réduit la charge de traitement que NetFlow fait peser sur les routeurs mais les volumes de trafic obtenus ne correspondent plus au volume du flot mais à une valeur estimée.

Une autre solution pour mesurer la demande de trafic OD consiste à exploiter les données SNMP. Le protocole SNMP (Simple Network Management Protocol) permet aux administrateurs de gérer les équipements et de diagnostiquer des problèmes réseau. SNMP se base sur un superviseur et sur des agents. Le superviseur permet aux administrateurs d'envoyer des requêtes aux agents pour récupérer des informations sur différents objets manageables. Ces objets manageables sont de type variés, informations matérielles, paramètres de configuration, et également des statistiques de performances sur les interfaces des équipements. Ces objets sont classés dans une base de données appelée MIB (Management Information Base) qui peut être consultée par les administrateurs grâce à SNMP pour recueillir les informations souhaitées.

Grâce au protocole SNMP très largement déployé il est possible pour les administrateurs d'un AS de recueillir à intervalles de temps réguliers les volumes de trafic sur chacune des interfaces des routeurs. Le volume de trafic sur une interface résulte de l'agrégation de tous les flots OD routés sur cette interface. En combinant les informations de routage à l'intérieur de l'AS et les volumes de trafic mesurés aux interfaces des routeurs grâce à SNMP on est en mesure de formaliser le problème d'estimation de la matrice de trafic

comme un problème linéaire inverse de la forme $Y = A\,X$. On note X la matrice de trafic OD, représentée par commodité sous la forme d'un vecteur colonne dont les éléments sont les demandes de trafic pour chaque couple Origine/Destination. On note Y un vecteur colonne dont les éléments représentent les volumes de trafic aux interfaces des différents routeurs du réseau. Et par ailleurs on note A la matrice de routage, matrice dont les éléments valent 0 ou 1, l'élément A_{ij} étant égal à 1 si le flot j est routé par l'interface i.

Matrice de trafic : état de l'art et contributions

Le système $Y = A\,X$ formalise un problème linéaire inverse puisque le vecteur Y et la matrice A sont connus, tandis que le vecteur X est l'inconnue du problème. Si N est le nombre de noeuds dans le réseau alors la taille de Y est de l'ordre de N^2 et la taille de X est proportionnel à N. Le système linéaire $Y = AX$ est donc un système linéaire inverse mal posé. Différentes approches ont été proposées pour traiter ce problème ce qui a donné lieu à de nombreuses publications sur le sujet [44, 39, 4, 5, 16, 35, 32, 50, 51]... Il existe également quelques tutoriaux [13]. Au printemps 2010 j'ai donné une vingtaine d'heures de cours sur la matrice de trafic à l'université de Pise dans le cadre du programme de formation doctorale. J'ai donné le même cours à l'automne 2010 à l'Institut d'Ingénierie Electrique (IIE) de l'Université de la République (UdelaR) en Uruguay où j'étais en séjour d'études.

L'article fondateur des travaux sur l'estimation de la matrice de trafic est celui de Vardi, 1996 [44]. Dans cet article Vardi fait l'hypothèse que les flots Origine/Destination X_i sont indépendants entre eux et distribués selon une loi de Poisson de paramètre λ_i inconnu. Sous l'hypothèse de Poisson la moyenne et la variance des flots Origine/Destination sont égaux à λ_i. La méthode de Vardi utilise les moyennes et covariances des volumes de trafic Y_j au niveau des interfaces et identifie une relation linéaire entre ces quantités et le vecteur λ qui représente la matrice de trafic moyenne. Comme le nombre de coefficients de covariance est de l'ordre de $N^2/2$ le problème linéaire inverse à résoudre est beaucoup mieux posé sous cette forme que le problème initial $\bar{Y} = A\,\lambda$. Vardi résoud ce problème LININPOS (*LINear INverse Problem with POSitivity constraints*) à l'aide d'un algorithme EM (*Expectation Maximization*), la possibilité d'utiliser l'algorithme EM pour résoudre des problèmes LININPOS ayant été identifiée dans un article précédent du même auteur (Vardi et Lee, 1993 [45]).

Peu de temps après Tebaldi et al. [39] ont publié un article sur l'utilisation de méthodes MCMC (*Markov Chain Monte Carlo*) pour estimer la matrice de trafic. L'approche algorithmique proposée se base sur une méthode de *Hastings-Metropolis within Gibbs*. Le principe est de combiner plusieurs algorithmes MCMC classiques, l'algorithme de Gibbs et l'algorithme de Hastings-Metropolis. Pour un cours introductif aux méthodes MCMC on peut lire par exemple le livre très pédagogique de Christian Robert [34]. Le principe proposé par Tebaldi est d'utiliser des techniques MCMC pour simuler la variable aléaatoire X (matrice de trafic) sous sa loi conditionnellement à la valeur de Y (volumes de trafic au niveau des interfaces des routeurs) en prenant en compte la relation $Y = A\,X$ et une loi a priori sur X.

Première contribution : méthode bayésienne itérative

Une limitation de la méthode proposée par Tebaldi [39] est que la loi a posteriori (loi de X sachant la valeur de Y) dépend fortement de la loi a priori sur X. Or la loi a priori sur X et en particulier les paramètres de cette loi sont inconnus. Dans nos articles [47] [48] [46] nous généralisons la méthode de Tebaldi au cas où la loi a priori sur X n'est pas connue. Nous supposons que chaque demande Origine/Destination X_i est distribuée selon un mélange de Gaussiennes. Nous proposons une méthode itérative qui combine deux étapes. Une étape reprend le même principe de *Hastings-Metropolis within Gibbs* que Tebaldi [39]. L'autre étape estime par un algorithme EM les paramètres (moyennes, variances, poids des composantes du mélange) du mélange de Gaussiennes qui modélise chaque $X_i(t)$ avec la particularité que les poids du mélange évoluent d'un instant t à un autre. Le nombre de composantes gaussiennes dans le mélange est estimé selon un critère de BIC. Les deux étapes sont répétées itérativement avec un échange approprié d'informations jusqu'à convergence de l'algorithme. Un principe inspiré du recuit simulé et de la façon dont les *informations extrinsèques* sont échangées dans le décodage itératif des turbo-codes convolutifs [3] [26] est appliqué pour permettre à l'algorithme de converger.

Cao et al. [4] exploitent comme Vardi l'idée d'une relation entre moyenne et variance des flots OD.

Dans l'article de Cao et al. cette relation est de type *loi de puissance* et les flots OD sont supposés être distribués selon une loi Gaussienne. Les paramètres des lois Gaussiennes sont ensuite estimés à l'aide d'un algorithme EM [1]. L'algorithme EM permet de résoudre de manière itérative des problèmes d'estimation paramétrique au sens du maximum de vraissemblance dans le cadre de modèles avec données partiellement observées. Le problème de la matrice de trafic entre dans ce cadre car seuls les volumes de trafic au niveau des liens Y sont mesurés, les volumes de trafic OD X_i n'étant pas observés directement. Dans [4] les auteurs font une hypothèse de stationnarité locale et estiment les paramètres de la matrice de trafic sur une fenêtre glissante. L'algorithme EM, très populaire dans le domaine du traitement de la parole, a également été utilisé dans le domaine de l'analyse de trafic, en particulier pour l'estimation des paramètres des modèles MMPP (Markov Modulated Poisson Process) [36].

Dans le cadre du réseau d'excellence européen EuroNGI [15] j'ai monté un cours doctoral de deux semaines sur les approches statistiques en modélisation de trafic (estimation paramétrique, théorie de la décision, grandes déviations, lois à queues lourdes, etc...). Les modèles de trafic du type processus de Poisson modulé Markov (MMPP) ont en particulier été étudiés dans le cadre de ce cours. Ce cours a été organisé à Télécom Bretagne en juin 2006 où il a réuni une trentaine de doctorants européens. Ce cours a été réitéré à l'université de Pise quelques mois plus tard dans le cadre de la formation doctorale des étudiants italiens.

Zhang et al. [50] [51] présentent une approche dite tomo-gravitaire du problème de la matrice de trafic. Le principe est de trouver la matrice de trafic la plus proche, au sens d'un critère issu de la théorie de l'information, d'une matrice estimée sous hypothèse d'indépendance entre les flots, tout en respectant les contraintes linéaires données par les volumes de trafic Y mesurés sur les interfaces des routeurs. Une méthode efficace de résolution de ce problème d'optimisation sous contraintes est proposée. Une comparaison des performances de différentes méthodes d'estimation est faite dans [27].

Castro et al. [10] introduisent l'idée d'utiliser une méthode de pseudo-vraissemblance pour estimer la matrice de trafic. L'idée est de modifier l'expression de la fonction de vraissemblance pour réduire le coût de calcul de l'estimation au sens du maximum de vraissemblance. L'estimation est alors faite grâce à une méthode dite pseudo-EM. Le problème initial est décomposé en une série de sous-problèmes, chacun correspondant à deux lignes particulières de la matrice de routage, et la fonction de vraissemblance initiale est remplacée par le produit des vraissemblances marginales de chaque sous-problème en ignorant les dépendances entre les sous-problèmes. Un premier intérêt de cette approche est qu'elle permet d'approximer l'estimateur du maximum de vraissemblance avec un coût de calcul significativement plus faible. Un deuxième intérêt est qu'elle peut se prêter à une implémentation distribuée, par exemple sur une architecture multi-coeur, sur une carte graphique ou sur une grille de calcul.

Seconde contribution : méthode simple basée sur la moyenne et la covariance des volumes sur les liens du réseau

Dans [28] [29] nous proposons une méthode très simple et à faible coût de calcul pour estimer la matrice de trafic. Ce travail a été développé en collaboration avec l'université technologique de Helsinki (HUT) dans le cadre de la thèse d'Ilmari Juva et d'une action de recherche spécifique (SRA, *Specific Research Activity*) que j'ai coordonnée au sein du réseau d'excellence européen EuroNGI. Cette SRA réunissait Télécom Bretagne, HUT (Helsinki), KTH (Stockolm), les universités de Rome 2 (Tor Vergata) et Roma 3 (La Sapienza) et France Télécom.

La contribution repose sur le constat suivant : si on ne prend en compte que les valeurs moyennes $\bar{Y}$ le problème $\bar{Y} = A\,\lambda$ est mal posé. Nous proposons de prendre en compte les valeurs des variances/covariances entre les Y_i sous l'hypothèse d'une relation fonctionnelle entre la moyenne et la variance des demandes Origine/Destination X_i pour obtenir un problème mieux conditionné. Nous proposons ensuite deux méthodes à faible coût de calcul pour estimer la matrice de trafic moyenne à partir des variances/covariances des Y_i, une méthode dite de projection et une méthode d'optimisation sous contraintes.

Des efforts ont été faits pour mettre à disposition de la communauté des chercheurs des données publiques sur la matrice de trafic. Il est difficile d'obtenir ces données des opérateurs ; elles sont jugées sensibles en raison de l'existence d'accords de peering et de transit qui définissent les conditions sur l'échange du trafic entre les systèmes autonomes. Les accords de peering sont stratégiques pour les opé-

rateurs ; on peut consulter à ce sujet le site de Bill Norton dit Dr Peering [14]. Un premier ensemble de matrices de trafic publiques correspond à des mesures obtenues à partir de données NetFlow sur le réseau Abilène en 2004 [49]. Un autre jeu de données publique a été produit dans le cadre du projet TOTEM (Université Catholique de Louvain, Université de Liège) à partir de données NetFlow de GEANT, réseau européen destiné à l'éducation et la recherche [37][43].

Les premiers travaux sur la matrice de trafic [44] [39] ont validé les algorithmes proposés sur des données de synthèse c'est-à-dire des matrices de trafic générées sous un modèle probabiliste. A partir du moment où des données publiques ont été disponibles, les travaux ont utilisé ces données publiques [49] [37] pour comparer les performances des différents algorithmes. L'objectif était de concevoir de nouveaux algorithmes d'estimation et de comparer leurs performances à ceux des techniques précédentes en utilisant les jeux de données publics.

Troisième contribution : calcul de la borne de Cramer-Rao pour le problème de la matrice de trafic

Cependant quelque soit la sophistication des méthodes envisagées il existe, du fait de la nature stochastique du problème, une borne sur la précision de l'estimation. La matrice d'information de Fisher quantifie l'information que les volumes de trafic mesurés Y apportent sur les paramètres à estimer. La borne de Cramer-Rao (CRLB, *Cramer Rao Lower Bound*), inverse de la matrice d'information de Fisher, donne une limite inférieure sur la variance asymptotique des estimateurs non biaisés. Dans [2] nous formalisons ce problème et montrons comment calculer la borne de Cramer-Rao pour le problème de la matrice de trafic. Le calcul de cette borne est très utile car il permet d'obtenir des intervalles de confiance quand la matrice de trafic est estimée au sens du Maximum de Vraissemblance (MLE, *Maximum Likelihood Estimate*). Cela permet également d'estimer la perte de performance de méthodes plus simples et moins calculatoires que le MLE comme celle que nous proposons dans [28] [29]. Le travail sur le calcul de la borne de Cramer Rao [2] a été réalisé avec Paola Bermolen de l'université de la République (Montévidéo, Uruguay). Cette collaboration a eu pour cadre un projet ECOS-Sud du Ministère des Affaires Etrangères auquel ont participé UdelaR (Uruguay), l'INRIA Rennes, l'université de Paris Sud (Orsay) et Télécom Bretagne.

Alors que les travaux précédents concernent l'estimation de la demande de trafic au niveau d'un système autonome (AS) certains auteurs se sont intéressés à l'évaluation de la demande de trafic au niveau inter-domaine. Feldmann et al. [17] proposent d'évaluer une matrice de trafic Web inter-domaine en exploitant les logs des CDN (*Content Delivery Networks*). Chang et al. [11] tentent de modéliser la demande de trafic inter-domaine en utilisant uniquement des données publiques. Ils proposent, à partir de données publiques, d'identifier les stratégies des AS (hébergement Web, accès résidentiel, business) puis d'inférer une matrice de trafic inter-AS sur la base d'un modèle gravitaire. L'évaluation de la demande de trafic a été étudiée dans d'autres contextes, par exemple pour des réseaux privés virtuels (*Virtual Private Network*, VPN) ou des réseaux optiques ou encore dans un contexte multi-services. Plus récemment les opérateurs de réseaux mobiles ont commencé à s'intéresser à l'estimation des mouvements d'usagers en exploitant les données de signalisation des réseaux cellulaires (handovers, charges des cellules, etc...). Les résultats de ces études exploitant les données de signalisation des réseaux de téléphonie mobile pour inférer les déplacements de population en zone urbaine intéressent les collectivités locales pour la planification de leur réseaux de transport. Des approches méthodologiques similaires à celles utilisées pour la matrice de trafic peuvent être employées pour traiter ce problème.

Après les travaux sur l'estimation de la matrice de trafic certains auteurs se sont intéressés à d'autres problématiques, en particulier la détection de changements brusques dans la demande de trafic ou encore la poursuite d'une matrice de trafic variable. Les changements brusques dans la demande de trafic intra-domaine s'expliquent majoritairement par des changements dans le routage inter-domaine et par l'interaction entre routage intra-domaine et routage inter-domaine [42] [41] [40].

Lakhina et al. [30] ont proposé une détection de changements basée sur une analyse en composantes principales (*Principal Components Analysis*, PCA). La décomposition PCA de la matrice de trafic a été introduite par les mêmes auteurs dans [31] . Le principe est de considérer que les premières composantes principales, celles qui captent le plus d'énergie, représentent le trafic "normal" tandis que les autres composantes représentent des anomalies potentielles. Les instabilités de cette approche ont été mises en évidence dans [33]. Les composantes principales sont apprises sur un jeu de données de mesures sur un

horizon de temps long (de l'ordre de la journée). Il est difficile d'obtenir sur cet horizon de temps des données indemnes de toute anomalie, de sorte que certaines anomalies sont intégrées dans les premières composantes qui sont supposées représenter le trafic "normal". C'est une limite classique des approches *data driven* où la base de décomposition est apprise sur les données à analyser elles-mêmes.

Quatrième contribution : détection/localisation de changement dans la matrice de trafic

Nous proposons au contraire de baser la détection de changements brusques sur une décomposition linéaire parcimonieuse de la matrice de trafic. Cette décomposition se fait sur une base de fonctions splines. Ce modèle linéaire parcimonieux permet par réjection de supprimer la matrice de trafic indemne d'anomalies du problème de détection. On considère alors la détection de changements dans la matrice de trafic comme un problème de détection d'anomalies avec paramètre de nuisance. Ce travail a été réalisé en collaboration avec l'université de technologie de Troyes (équipe d'Igor Nikiforov) dans le cadre du post-doctorat de Lionel Fillatre. Nous appliquons des algorithmes de détection et localisation d'anomalies dont les propriétés d'optimalité sont démontrées (Fouladirad et Nikiforov, [23]) et appliquées à un contexte de détection d'anomalies en tomographie par rayons X (Fillatre et Nikiforov, [19]). Différentes techniques de détection/localisation d'anomalies avec paramètres de nuisance sont appliquées, avec l'avantage par rapport aux techniques précédentes, d'avoir des propriétés d'optimalité solides en termes de délai de détection, taux de fausse alarme, taux de fausse localisation. En pratique les performances observées par simulation sur les trois jeux de données de test (Abilène [49], GEANT [37], données France Télécom) sont bien meilleures que celles des travaux précédents [30] ce qui vérifie en pratique les bonnes propriétés théoriques des algorithmes proposés. Ces travaux ont été publiés dans différentes conférences [21] [22] [20] [8] et dans un article de journal [9]. Le travail initié dans le cadre du post-doctorat de Lionel Fillatre a été poursuivi par la thèse de Pedro Casas et complété par une étude sur le routage en présence d'incertitude dans la demande de trafic (changements brusques, erreur d'estimation, variabilité journalière, etc...).

Cinquième contribution : estimation/poursuite de la matrice de trafic par filtrage de Kalman

Soule et al. [38] proposent de poursuivre les variations de la matrice de trafic par une technique de filtrage de Kalman ; un changement brusque dans la demande de trafic est détecté lorsque la valeur de l'innovation dans l'algorithme de Kalman dépasse un certain seuil. Dans [7] nous mettons en évidence une limitation de cet article. Nous remarquons que le modèle linéaire Gaussien d'évolution de la matrice de trafic de [38] implique nécessairement que la matrice de trafic soit de moyenne nulle. Les équations du filtrage de Kalman sont généralement énoncées sous l'hypothèse de processus linéaires gaussiens centrés. Dans le cas de la matrice de trafic la moyenne du processus est clairement non nulle. En raison de ce problème de modélisation, un terme d'erreur est propagé dans les différentes étapes de la boucle de Kalman (filtrage, prédiction) et cette erreur soit converge vers une constante non nulle soit diverge vers l'infini. Dans [7] nous analysons les implications de cette erreur de modélisation puis nous proposons des corrections simples au modèle d'évolution de la matrice de trafic pour résoudre ce problème. Nous considérons deux cas : une matrice de trafic de moyenne fixe (non nulle), et une matrice de trafic de moyenne variable. Ces corrections au modèle permettent d'obtenir un algorithme beaucoup plus stable et précis que [38].

La matrice de trafic est variable soit en raison de changements brusques en général liés à des modifications des annonces du routage externe BGP (*Border Gateway Protocol*) soit en raison d'une saisonnalité journalière et hebdomadaire de la demande de trafic liée aux variations de l'activité humaine sur le réseau Internet. Il est important de construire des modèles de trafic permettant de rendre compte de la saisonnalité naturelle de la demande. Dans [18] nous proposons d'utiliser des modèles de série temporelle SARIMA (*Seasonal Auto Regressive Integrated Moving Average*) pour représenter la saisonnalité du trafic. Une représentation sous forme de modèle d'état linéaire permet d'utiliser un filtrage de Kalman pour prédire les évolutions régulières du trafic sur plusieurs jours avec une grande précision.

Sixième contribution : estimation de la matrice de trafic par réseau de neurones stochastique

Les techniques de réseaux de neurones donnent de très bons résultats dans un grand nombre de problèmes d'inférence avec des applications très variés. Dans [6] nous proposons d'utiliser une méthode d'apprentissage, les réseaux de neurones aléatoires (RNN, *Random Neural Networks* [24]) pour estimer

la valeur de la matrice de trafic X à partir des volumes de trafic Y sur les interfaces des routeurs. Les performances obtenues par la méthode RNN sont très bonnes mais elles nécessitent une calibration du réseau de neurones et cette calibration suppose d'avoir accès à une mesure directe de la matrice de trafic, par exemple à partir des données NetFlow, pendant un certain temps.

QUELQUES PUBLICATIONS INCLUES DANS LE MANUSCRIT

VATON Sandrine, BEDO J.S, GRAVEY Annie, Advanced Methods for the Estimation of the Origin Destination traffic matrix. Publié dans Performance Evaluation and Planning Methods for the Next Generation Internet, Editors : A. Girard, B. Sanso, F. Vazquez-Abad, 2005

Cet article présente un état de l'art détaillé sur les méthodes de résolution du problème de la matrice de trafic : méthode de pénalisation, méthode basée sur les moments du 1er et du 2nd ordre, méthode tomo-gravitaire, méthode EM, méthode MCMC. Puis dans un second temps nous proposons une amélioration de la méthode MCMC de Tebaldi et al. Nous identifions une faiblesse de cette méthode : le choix arbitraire de la loi a priori sur chaque flot Origine/Destination. Puis nous présentons une technique itérative permettant de calibrer ces lois a priori. Cette technique ne nécessite pas de mesurer directement les flots Origine/Destination ; au contraire elle n'exploite que les données du problème, à savoir les volumes de trafic agrégés sur les liens du réseau (données SNMP) et le routage. La méthode proposée se base sur un échange itératif d'informations entre différents blocs : un bloc de calibration d'un modèle de mélanges de Gaussiennes sur chacun des couples Origine/Destination, et un bloc d'estimation de la matrice de trafic selon la méthode MCMC de Tebaldi et al. L'échange itératif d'informations s'inspire du décodage itératif des codes convolutifs concaténés en parallèle (turbo-codes convolutifs). Les performances de la méthode sont comparées à celles de l'état de l'art ; on montre que la technique proposée améliore nettement les performances de la méthode de Tebaldi et al., et obtient de meilleurs résultats que toutes les méthodes de la littérature. En particulier elle permet une bien meilleure estimation des *outliers* (valeurs atypiques de certains flots à certains instants) et des petits flots qui sont classiquement mal estimés car ils servent en règle générale de variable d'ajustement dans le problème.

JUVA Ilmari, VATON Sandrine, VIRTAMO Jorma Quick traffic matrix estimation based on link count covariances. ICC 2006 : IEEE International Conference on Communications, June 11-15, Istanbul, 2006

Le travail présenté dans cet article a été fait dans le cadre de la thèse de Ilmari JUVA à HUT (Helsinki University of Technology). La collaboration avec HUT a été initiée dans EuroNGI et en particulier dans le cadre d'une activité de recherche spécifique sur la matrice de trafic que j'ai coordonnée. L'article s'intéresse à l'estimation de la matrice de trafic moyenne λ à partir des volumes de trafic sur les liens du réseau. Nous proposons d'utiliser les moyennes et covariance des volumes de trafic mesurés sur les liens du réseau comme variables d'entrée du problème. Nous faisons l'hypothèse d'une relation fonctionnelle (de type loi de puissance) entre la moyenne et la variance des volumes de trafic OD. Sous cette hypothèse le problème d'estimation de λ devient un problème inverse bien posé. Nous proposons deux approches à faible coût de calcul pour estimer la matrice de trafic moyenne λ.

BERMOLEN Paola, VATON Sandrine, JUVA Ilmari Search for optimality in traffic matrix estimation : a rational approach by Cramer-Rao lower bounds. NGI'06 : 2nd Conference on Next Generation Internet Design and Engineering, 3-5 april ,Valencia, Spain, 2006, pp. 224-231

Cet article présente les résultats d'une collaboration avec la faculté d'ingénierie d'UdelaR (Université de la République, Uruguay). Le travail a été effectué lors du séjour à Télécom Bretagne de Paola Bermolen,

jeune chercheuse en mathématiques à l'UdelaR, dans le cadre d'un projet ECOS-Sud.

L'idée de cet article part de la constatation suivante. En règle générale les auteurs qui ont travaillé sur le problème de la matrice de trafic proposent une méthode puis comparent ses performances par simulation avec celles des autres méthodes. Cependant d'un point de vue statistique et théorie de l'information le problème de la matrice de trafic est un problème avec données manquantes. Les volumes OD ne sont pas mesurés directement ; seuls les volumes agrégés sur les liens du réseau (données SNMP) sont mesurés, ce qui induit une perte irrémédiable d'information. Aucun estimateur ne peut être "parfait" ; par la nature du problème tous les estimateurs seront entachés d'erreurs.

Il existe cependant une borne sur la qualité que l'on peut espérer obtenir d'un estimateur. Cette borne s'appelle la borne de Cramer-Rao (CRLB, *Cramer-Rao Lower Bound*), qui est l'inverse de la matrice d'information de Fisher. Notre article fournit une expression analytique de la borne de Cramer-Rao pour le problème de la matrice de trafic. Le calcul est fait sous une hypothèse Gaussienne sur les volumes de trafic OD et sous l'hypothèse d'une relation fonctionnelle de type loi de puissance entre la moyenne et la variance des flots OD.

Après avoir établi l'expression analytique de la borne de Cramer-Rao pour notre problème nous proposons deux applications pour ce calcul. D'une part le fait de disposer d'une expression analytique de la borne permet de quantifier la perte de performances d'une méthode simple par rapport à une méthode asymptotiquement optimale mais de complexité plus importante basée sur le critère du maximum de vraissemblance. La deuxième application est plus pratique. Alors que SNMP est un protocole largement déployé, à l'époque de ces travaux la mise en place de routeurs NetFlow représentait un facteur de coût non négligeable pour les opérateurs. Nous considérons dans cet article la détermination du placement optimal de quelques points de capture NetFlow dans un réseau. Ces points de capture permettent de mesurer directement certains flots OD. Les autres flots doivent être estimés à partir des mesures SNMP. Le placement préconisé des points de capture NetFlow est celui qui permet de maximiser l'information de Fisher.

CASAS HERNANDEZ Pedro, VATON Sandrine, FILLATRE Lionel, NIKIFOROV Igor, Optimal Volume Anomaly Detection and Isolation in Large-Scale IP Networks using Coarse-Grained Measurements, Computer Networks (COMNET), volume 54, issue 11, pages 1750-1766, august 2010

Cet article présente des travaux réalisés lors du séjour post-doctoral de Lionel Fillatre et de la thèse de doctorat de Pedro Casas. Nous présentons des algorithmes de détection/isolation de changements dans la demande de trafic OD sur un système autonome. Contrairement aux méthodes de la littérature, les algorithmes que nous proposons ont des propriétés d'optimalité bien démontrées ; ces propriétés d'optimalité s'expriment en termes de taux de fausse alarme et de non détection (approche *snapshot*), ou en termes de délai de détection/isolation et de taux de fausse alarme ou fausse isolation (approche séquentielle). L'utilisation d'algorithmes ayant des propriétés d'optimalité bien démontrées permet de s'assurer du bon comportement de la méthode indépendamment d'une évaluation particulière.

Nous proposons une décomposition spatiale linéaire et parcimonieuse de la matrice de trafic sur une base de splines. Ce modèle linéaire parcimonieux nous permet de traiter la détection de changements comme un test statistique en présence de paramètres de nuisance. Les paramètres de nuisance représentent la matrice de trafic habituelle, décomposable sur une base de fonctions splines. La matrice de trafic normale est éliminée du problème de décision par réjection et les changements sont détectés dans les résidus. Après avoir introduit le modèle linéaire parcimonieux basé splines et l'avoir validé sur différents jeux de données (réseaux Abilène, GEANT et France Télécom OTIP) nous proposons divers algorithmes.

Le premier algorithme fournit un estimateur de la demande de trafic Origine/Destination ; il s'agit d'un estimateur au sens du maximum de vraissemblance se basant sur le modèle introduit précédemment . Le second algorithme permet la détection d'une anomalie dans la demande de trafic Origine/Destination selon une approche *snapshot*, c'est-à-dire que la décision se base sur une seule fenêtre temporelle. A chaque nouvelle mesure SNMP le test décide entre deux hypothèses H_0 (trafic conforme au modèle) et H_1 (trafic non conforme). Le test proposé est celui qui offre les meilleures performances ; c'est un test UBCP (*Uniformly Best Constant Power*) au sens de Wald. Cette propriété d'optimalité théorique se vérifie bien sûr en pratique par des performances nettement meilleures que celles de la littérature sur les

trois jeux de données considérés. Enfin le dernier algorithme permet la détection et la localisation d'un changement dans la demande de trafic selon une approche séquentielle. L'algorithme a la capacité de localiser l'anomalie c'est-à-dire de déterminer quel flot OD particulier a subi un changement significatif (augmentation ou baisse de la demande). Il s'agit d'un algorithme séquentiel ; à chaque fenêtre temporelle les nouvelles mesures SNMP sont analysées et la décision est prise de lever ou non une alarme, la décision se basant sur tout le passé. L'algorithme proposé est optimal au sens où il minimise le délai moyen de détection/localisation sous contraintes de taux de fausse alarme et de fausse localisation bornés supérieurement.

CASAS HERNANDEZ Pedro, VATON Sandrine, CHONAVEL Thierry, Efficient Methods for Traffic Matrix Modelling and On-Line Estimation in Large Scale IP Networks, ITC'21 : 21st International Teletraffic Congress, septembre 2009, Paris.

Cet article s'intéresse à la poursuite des variations de la demande de trafic Origine/Destination par filtrage de Kalman. Dans le problème de la matrice de trafic le modèle d'observation $Y = R\,X$ est par nature linéaire. Il est également possible de faire l'hypothèse d'un modèle d'état linéaire, par exemple un AR(1) ; le modèle d'état caractérise l'évolution temporelle de la demande de trafic. On est donc dans un cadre où les techniques de filtrage de Kalman pour la poursuite des variations temporelles de la demande sont adaptées puisque les modèles d'état et d'observation sont linéaires et éventuellement gaussiens.

Dans notre article nous mettons en évidence une limitation d'un article de la littérature [38]. Dans celui-ci les auteurs font l'hypothèse d'un modèle d'état AR(1) pour caractériser l'évolution de la demande de trafic. Ce modèle d'état est de la forme $X(t+1) = A\,X(t) + N(t)$ où $N(t)$ est un bruit blanc gaussien centré. Puis les auteurs appliquent les équations classiques de Kalman (boucle de Kalman) pour poursuivre/prédire les évolutions de la demande au vu des mesures $Y(t)$.

Les auteurs considèrent deux cas : une matrice d'évolution A diagonale (cas de flots Origine Destination indépendants entre eux) et une matrice A non diagonale. Ils constatent qu'une re-calibration régulière de la matrice A est nécessaire. On peut observer sur une des figures un écart entre le filtre de Kalman et la valeur réelle de la demande, ce qui laisse penser que le filtre ne converge pas. L'argument des auteurs est qu'ils ne s'intéressent pas à l'estimation de la demande mais à la détection d'un changement brusque dans la demande, et qu'un écart semble-t-il constant entre l'estimateur et la valeur réelle de la demande n'est pas vraiment gênant.

Nous analysons dans notre article que le modèle d'état de la forme $X(t+1) = A\,X(t) + N(t)$ où $N(t)$ est centré impose que la matrice de trafic moyenne λ soit dans le noyau de $I - A$ où I est la matrice identité. En particulier dans le cas où A est diagonale cela implique $\lambda = 0$ ce qui est manifestement une hypothèse erronée. Dans le cas où A est non diagonale la détermination d'une matrice A telle que λ vérifie $(I - A)\lambda = 0$ explique certainement pourquoi des re-calibrations fréquentes de A sont rendues nécessaires par les variations de λ.

Dans un premier temps nous analysons les implications de ce problème de modélisation. Nous démontrons que si A est diagonale et si les équations de la boucle de Kalman sont appliquées alors l'erreur d'estimation soit diverge vers l'infini soit converge vers une valeur fixe non nulle. Cela est cohérent avec l'écart observé sur une des figures de [38] entre le filtre et la valeur réelle de la matrice de trafic.

Puis dans un second temps nous montrons comment résoudre ce problème. Nous intégrons la valeur non nulle de la demande de trafic moyenne λ dans la définition de l'état du système en introduisant un état étendu sous la forme $U(t) = \left[X(t)^T, \lambda^T\right]^T$ (T est l'opérateur de transposition). Sous cette forme les équations classiques du filtrage de Kalman peuvent être appliquées pour estimer la matrice de trafic moyenne λ et les variations $X(t)$ autour de cette moyenne. Nous considérons deux cas : un cas stationnaire avec λ constant, et un cas non stationnaire dans lequel on autorise des variations lentes sur la matrice de trafic moyenne λ. Les simulations montrent que notre proposition corrige les limitations de l'article [38] et en particulier les problèmes de convergence, et que sous cette forme le filtre de Kalman est un très bon estimateur de la demande de trafic OD.

PERSPECTIVES

Comme je l'ai dit dans l'introduction la solution déployée aujourd'hui par les opérateurs pour estimer la demande de trafic se base sur la récolte de statistiques de trafic au niveau flot avec des outils comme NetFlow. L'inférence de la matrice de trafic à partir de données SNMP et les problèmes liés à la résolution de ce problème mathématique mal posé est devenu un domaine de recherche beaucoup moins actif qu'il y a une dizaine d'années.

Les travaux que nous avons présentés dans ce manuscrit constituent une contribution essentiellement méthodologique sur différents aspects de ce problème théorique. Ces aspects concernent l'inférence, la poursuite ou encore la détection et la localisation de changements brusques dans un cadre où les données qui nous intéressent (matrice de trafic) ne sont pas observées directement mais au travers d'une transformation linéaire avec peu d'observations (mesures SNMP) et un grand nombre de variables non observées.

Sur le plan des méthodes de traitement du signal il serait possible d'approfondir certains aspects du travail pour améliorer nos techniques d'estimation. Par exemple nous avons commencé à réfléchir sur l'estimation des paramètres du modèle d'état et du modèle d'observation linéaires gaussien sur lesquels se base l'estimation, la poursuite et la prédiction de la matrice de trafic par filtrage de Kalman. Il est possible d'estimer les paramètres de ce modèle par un algorithme EM (*Expectation Maximization*) classiquement utilisé pour les problèmes d'estimation paramétrique dans les modèles avec données manquantes. Ce travail sur l'estimation des paramètres du modèle linéaire gaussien vient compléter notre étude publiée dans [7].

Le problème de tomographie réseau c'est-à-dire l'inférence de paramètres de QoS comme le délai ou les taux de perte sur les liens internes d'un réseau à partir de mesures de bout-en-bout a beaucoup de points communs, du point de vue de sa formalisation, avec le problème de la matrice de trafic. L'expérience acquise sur la matrice de trafic peut être exploitée pour traiter des problèmes relatifs à la tomographie du réseau qui est un domaine de recherche toujours actif [25].

Références

[1] A.P.Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B (JRSS-B)*, 39 :1–38, 1977.

[2] P. Bermolen, S. Vaton, and I. Juva. Search for optimality in traffic matrix estimation : a rational approach by Cramer-Rao lower bounds. In *NGI'06 : 2nd Conference on Next Generation Internet Design and Engineering*, 2006.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding : turbo-codes. In *IEEE ICC '93, Geneva*, pages 1064 – 1070, 1993.

[4] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-Varying Network Tomography : Router Link Data. *Journal of the American Statistical Association*, 95(452), Dec. 2000.

[5] J. Cao, S. Vander Wiel, B. Yu, and Z. Zhu. A Scalable Method for Estimating Network Traffic Matrices from Link Counts. Technical report, Bell Labs, 2000.

[6] P. Casas and S. Vaton. On the use of random neural networks for traffic matrix estimation in large-scale IP networks. In *Proceedings of the 1st Int. Workshop on Traffic Analysis and Classification*, IWCMC'10, 2010.

[7] P. Casas, S. Vaton, L. Fillatre, and T. Chonavel. Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks. In *ITC'21 : 21st conference on International Teletraffic Congress*, 2009.

[8] P. Casas, S. Vaton, L. Fillatre, and I. Nikiforov. Volume anomaly detection in data networks : an optimal volume detection vs the PCA approach. In *FitraMen'08 : International Workshop on Traffic Management and Traffic Engineering for the Future Internet*, 2008.

[9] P. Casas, S. Vaton, L. Fillatre, and I. Nikiforov. Optimal volume anomaly detection and isolation in large-scale ip networks using coarse-grained measurements. *Computer Networks*, 54(11) :1750–1766, 2010.

[10] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography : recent developments. *Statistical Science*, 19 :499–517, 2004.

[11] H. Chang, S. Jamin, Z. Morley Mao, and W. Willinger. An empirical approach to modeling inter-as traffic matrices. In *Internet Measurment Conference*, 2005.

[12] CISCO. CISCO IOS NetFlow. `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`.

[13] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet Tomography. *IEEE Signal Processing Magazine*, 19 :47–65, 2002.

[14] Dr Peering (Bill Norton). DrPeering.net : The tools and talent needed to make strategic peering decisions. . `http://drpeering.net/`.

[15] European Commission, Information Society Technologies, Sixth Framework Program. EuroNGI Network of Excellence. `http://eurongi.enst.fr/en_accueil.html`.

[16] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks : methodology and experience. *SIGCOMM Comput. Commun. Rev.*, 30 :257–270, 2000.

[17] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. De Prisco, and R. Sundaram. A methodology for estimating interdomain web traffic demand. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, 2004.

[18] L. Fillatre, D. Marakov, and S. Vaton. Forecasting seasonal traffic flows. In *EuroNGI Workshop on QoS and Traffic Control*, 2003.

[19] L. Fillatre and I. Nikiforov. Non-bayesian detection and detectability of anomalies from a few noisy tomographic projections. *IEEE Transactions on Signal Processing*, 55(2) :401–413, 2007.

[20] L. Fillatre, I. Nikiforov, P. Casas, and S. Vaton. Optimal volume anomaly detection in network traffic flows. In *EUSIPCO'08 : 16th European Signal Processing Conference*, 2008.

[21] L. Fillatre, I. Nikiforov, and S. Vaton. Détection-localisation séquentielle d'anomalies volumiques dans un réseau. In *GRETSI 2007 : 21ème collloque sur le traitement du signal et des images, 11-14 septembre, Troyes, France*, 2007.

[22] L. Fillatre, I. Nikiforov, S. Vaton, and P. Casas. Sequential non Bayesian network traffic flows anomaly detection and isolation. In *IWAP 2008 : International Workshop on Applied Probability*, 2008.

[23] M. Fouladirad and I. Nikiforov. Optimal statistical fault detection with nuisance parameters. *Automatica*, 41(7) :1157–1171, 2005.

[24] E. Gelenbe. Random neural networks with negative and positive signals and product form solution. *Neural Computation*, 1 :502–511, 1989.

[25] D. Ghita, P. Thiran, and A. Argyraki (directeurs). *Practical Network Tomography.* Thèse de doctorat de l'Ecole Polytechnique Fédérale de Lausanne, EPFL, Lausanne, 2012.

[26] A. Glavieux, P. Adde, G. Battail, E. Biglieri, M. Jezequel, A. Poli, S. Vaton, R. Pyndiah, A. Picart, C. Douillard, and C. Berrou. *Codage de canal - des bases théoriques aux turbocodes (sous la direction de Alain GLAVIEUX).* Traité IC2 : Traitement du signal et de l'image. Hermès Science / Lavoisier, Paris, 2005.

[27] A. Gunnar, M. Johansson, and T. Telkamp. Traffic matrix estimation on a large ip backbone : a comparison on real data. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, 2004.

[28] I. Juva, S. Vaton, and J. Virtamo. Quick traffic matrix estimation based on link counts covariances. In *2nd EuroNGI Workshop On Traffic Engineering, Protection and Restoration for Next Generation Internet*, 2005.

[29] I. Juva, S. Vaton, and J. Virtamo. Quick traffic matrix estimation based on link count covariances. In *ICC 2006 : IEEE International Conference on Communications*, 2006.

[30] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *In ACM SIGCOMM*, pages 219–230, 2004.

[31] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proceedings of the joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '04/Performance '04, 2004.

[32] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation : existing techniques and new directions. *SIGCOMM Comput. Commun. Rev.*, 32 :161–174, 2002.

[33] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. In *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '07, 2007.

[34] C. Robert. *Méthodes de Monte Carlo par Chaînes de Markov.* Economica, 1996.

[35] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in Measuring Backbone Traffic Variability : Models, Metrics, Measurements and Meaning. In *Internet Measurement Workshop*, Marseille, Nov. 2002.

[36] Tobias Rydén. An em algorithm for estimation in markov-modulated poisson processes. *Comput. Stat. Data Anal.*, 21 :431–447, 1996.

[37] S. Uhlig. The TOTEM Project Datasets. `http://totem.info.ucl.ac.be/dataset.html`.

[38] A. Soule, K. Salamatian, A. Nucci, and N. Taft. Traffic matrix tracking using kalman filtering. In *In ACM SIGMETRICS Performance Evaluation Review (PER*, 2005.

[39] C. Tebaldi and M. West. Bayesian Inference on Network Traffic Using Link Count Data. *Journal of the American Statistical Association*, 93(442), June 1998.

[40] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded : Impact of routing changes. In *Passive and Active Network Measurement*, volume 3431 of *Lecture Notes in Computer Science*, pages 251–264. 2005.

[41] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in ip networks. In *SIGMETRICS*, pages 307–319, 2004.

[42] R. Teixeira, A. Shaikh, T. Griffin, and G. M. Voelker. Network sensitivity to hot-potato disruptions. In *SIGCOMM*, pages 231–244, 2004.

[43] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev.*, 36 :83–86, January 2006.

[44] Y. Vardi. Network Tomography : Estimating Source-Destination Traffic Intensities From Link Data. *Journal of the American Statistical Association*, 91(433), March 1996.

[45] Y. Vardi and D. Lee. From image deblurring to optimal investments : maximum likelihood solutions for positive linear inverse problems. *Journal of the Royal Statistical Society Series B (JRSS-B)*, 55 :569–612, 1993.

[46] S. Vaton, J.S Bedo, and A. Gravey. *Advanced methods for the estimation of the origin destination traffic matrix*, volume XVI, chapter Performance Evaluation and Planning Methods for the Next Generation Internet (25 th du GERARD). A. Girard, B. Sanso, F. Vazquez-Abad, 2005.

[47] S. Vaton and A. Gravey. Iterative Bayesian Analysis of Network Traffic Matrices in the Case of Bursty Flows. In *Internet Measurement Workshop*, Marseille, Nov. 2002.

[48] S. Vaton and A. Gravey. Network Tomography : an Iterative Bayesian Analysis. In *ITC 18th*, Berlin, Aug.-Sep. 2003.

[49] Y. Zhang. Six months of Abilene traffic matrices. `http://www.cs.utexas.edu/~yzhang/research/AbileneTM/`.

[50] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads. In *ACM SIGMETRICS*, San Diego, Jun. 2003.

[51] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho. Estimating point-to-point and point-to-multipoint traffic matrices : an information-theoretic approach. *IEEE/ACM Trans. Netw.*, 13 :947–960, October 2005.

Advanced methods for the estimation of the Origin Destination traffic matrix

S. Vaton [*], J.S. Bedo [†], A.Gravey [‡]

January 21, 2005

Abstract

For lots of traffic engineering tasks, telecommunications operators need good knowledge about the traffic which transit through their networks. This information is fully represented by the matrix of the volumes of data which go from any entry node to any exit node during a period of time. This matrix is called the origin-destination (OD) traffic matrix. However such a matrix is not directly available. Only measures of the volumes of data which transit through a link between routers can be obtained easily with the help of Simple Network Management Protocol (SNMP). These measures are called link counts.

Lots of techniques have been proposed to estimate the traffic matrix from the link counts. Among those, statistical methods propose to model the demand for each OD pair in order to chose a possible traffic matrix which fits the reality of networks. Nevertheless, the model is often arbitrary and don't take into account the temporal dimension of traffic. In this paper, we claim that a temporal model of the traffic for each OD pair could be trained from the link counts only. We prove the validity of our approach on a one router network on which direct measurements of the OD counts were made available. Then, we compare our results to other methods and show that their accuracy is the best.

1 Introduction

With the diversification of network applications, the volume of data carried on international backbones is increasingly sporadic and unpredictable. As a consequence, traffic engineering, dimensioning and routing are outstanding issues. A good knowledge of traffic behavior is necessary for an efficient deployement of traffic engineering tools.

To improve their quality of service (QoS) while reducing their costs, operators must find ways to measure and predict the traffic offered by their customers and peers. That is to say, they need to know the volume of traffic which transits from any edge node to any other edge node of their own network, where an edge node is a point where traffic enters and/or exits the operator's network (i.e. a router or a point of presence (POP)). This volume expressed in a number of bytes or packets observed during a given period of time (typically 5 or 10 minutes) is called origin-destination (OD) counts. These OD counts are usually represented in the form of a matrix where lines represent origin nodes and columns represent destination nodes. This $N \times N$ matrix where N is the number of edge nodes connected to the network is called the OD traffic matrix.

Unfortunately, the OD traffic matrix cannot usually be measured directly on large commercial backbones. Indeed, commercial software e.g. CISCO Netflow first sample the packets transiting through a given router and then infer their origin and destination by analyzing their headers. Using this method presents several

[*]ENST Bretagne, Brest, France. *Ce programme reçoit le soutien financier de la Région Bretagne, programme 1042, opération A3C942.*

[†]student, Ecole Polytechnique, Palaiseau, France.

[‡]ENST Bretagne, Brest, France.

drawbacks. Firstly, it requires deploying the same software on all core routers which may not be feasible due to cost and heterogeneity of networking equipment. Secondly, such a software generates a significant CPU demand on the router that may negatively impact on the global router performance. Thirdly, synchronizing individual measures and storing measurement data is not simple. Finally, these methods provide no guarantee that the sampling procedure provides a good representation of the real OD traffic matrix due to the heterogeneity of traffic demands.

On the other hand, SNMP (*Simple Network Management Protocol*) counters routinely provide global counts of traffic aggregates observed on router interfaces. These are naturally called link counts since they correspond to the amounts of traffic transiting on a link between two routers. Furthermore, the network operator is obviously aware of its routing policy and may easily infer the paths followed by packets between origins and destinations. The routing information can be summed up in terms of another matrix, the routing matrix, where each line corresponds to a link in the network, and each column is an OD pair. A typical routing matrix element is a weight between 0 and 1 that represents the probability that a packet sent between the origin and destination transits through this link. For example, a weight of 1 (respectively 0.5) means that all (respectively half of) the traffic of the OD pair transits through this particular link. In this paper, we address the issue of inferring the OD matrix from the above easily obtained information, i.e. links volume and routing matrix.

The traffic matrix problem is typically an ill-posed problem. If $\mathbf{y}$ denotes the column vector of the link counts and $\mathbf{x}$ denotes the column vector of the OD counts then

$$\mathbf{y} = \mathbf{A}\ \mathbf{x} \tag{1}$$

where $\mathbf{A}$ is the routing matrix.

Note that $\mathbf{A}$ is not invertible and that the linear system is actually underdetermined. Indeed, the number r of link counts is usually much smaller than the number c of OD pairs; typically, if N is the number of edge nodes, then r is of the same order of magnitude as N, whereas c is of the order of N^2, where N may be as large as 200. This implies that the number of solutions to the above linear system is infinite. To be more precise, the dimension of the solutions state space is $c - r$.

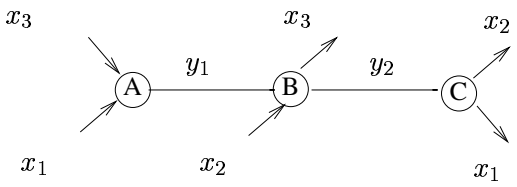


Figure 1: A simple example network with 2 monodirectional links and 3 OD pairs.

To illustrate this issue, consider the example of a very simple network shown in Figure 1 with two links and three OD pairs. Obviously $y_1 = x_1 + x_3$ and $y_2 = x_1 + x_2$. Let us suppose for example that $y_1 = 3$ and $y_2 = 5$. In this case, there are four integer positive solutions (and even more non-integer solutions) : $(x_1, x_2, x_3) = (0, 5, 3), (1, 4, 2), (2, 3, 1)$ and $(3, 2, 0)$. These solutions have nothing in common, except that they all satisfy the equations $x_1 + x_3 = 3$ and $x_1 + x_2 = 5$.

Solving the problem of inferring the OD matrix $\mathbf{x}$ from links volume $\mathbf{y}$ and routing matrix $\mathbf{A}$ thus consists of selecting one "good" solution among the ones which satisfy Equation (1).

The rest of the paper will be organised as follows. In Section 2, we will present a general taxonomy of existing methods for the traffic matrix problem. Then, we will present two non-statistical methods : penalization method in Section 3 and gravity models in Section 4. In Section 5, we will begin our exploration of statistical methods for the traffic matrix problem with the Expectation Maximization algorithm. Section 6 is a detailed tutorial on Markov Chain Monte Carlo (MCMC) techniques applied to the estimation of the

OD traffic matrix. Then comes the main Section 7 which describes our own algorithm for traffic matrix estimation. It consists of an improvement over the MCMC technique. Last but not least, we show in Section 8 the numerical results we obtained on datasets from a real network and compare the performance of our algorithm with others. Section 9 is a conclusion.

2 A taxonomy of OD matrix estimation methods

As we saw in the introduction, the OD matrix can be estimated from the link values as a solution of the linear system $\mathbf{y} = \mathbf{A}\mathbf{x}$. As this system is underdetermined, there is an infinite number of solutions and one, or a selection, of solutions of that system must be selected as better than others. To do so, a variety of methods have been considered in the literature. Most of the time, these methods have been validated on synthetic datasets produced with different modelling assumptions (Poisson, Gaussian, etc...). Some of these methods have also been compared on measured backbone traffic [1] or on LAN traffic [2]. In some cases, cross validation for some OD flows was made possible by running special software (for example CISCO *Netflow*) on a few routers. In other cases, a validation was done on synthetic traffic.

In this Section, we orientate the reader through the variety of existing and other possible techniques. Various criteria can be considered to classify existing OD matrix estimation methods, as well as some other methods that have not yet been published but whose study could be of interest.

2.1 Single OD matrix versus set of OD matrices

OD traffic matrix is an old issue in dimensioning telephone networks. Classically, to dimension a circuit switched telephone network, only the mean of each OD flow is considered. This results in the definition of one *single* OD traffic matrix. The rationale for doing so is probably the Poisson modelling of telephone networks, since the distribution of a Poisson random variable is completely specified by its mean λ. Since the 1990's it has nevertheless been recognized that data network traffic is not consistent with the Poisson model [3]. Internet data traffic has indeed properties such as fractal or multi-fractal behaviour, self-similarity at different timescales, heavy tails, etc... It is recognized that the Poisson model would result in serious failures in modelling, predicting performances, and enabling relevant network design.

In practice, on a data network, SNMP measurements routinely provide links volume at a rate as high as one SNMP request to the router MIB (*Management Information Base*) every 5 minutes. This provides up to 12 OD matrices per hour, 288 per day, 2016 per week ... Each of these OD matrices $\mathbf{x}_t, t = 1, \ldots, T$ is a solution of the linear system $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ where t refers to a given period of time of typical duration 5 or 10 minutes. This set of traffic matrices can then be mapped into a set of constraints for traffic engineering. Taking into account this set of constraints rather than a single average OD matrix will result in a more accurate traffic engineering than a simple overdimensioning based on the average OD matrix would do.

2.2 Statistical versus non-statistical methods

As mentioned previously, SNMP measurements routinely provide link information $\mathbf{y}_t$, and the OD matrix $\mathbf{x}_t$ must be estimated with the constraint that $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$, or $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{n}_t$ if some measurement noise (due to desynchronization of SNMP and/or IGP measurements for example) is incorporated into the model. As the routing matrix $\mathbf{A}$ is not invertible, some additional criteria are needed to select one or a set of solutions as better than others.

To do so, it is possible to consider the OD matrix $\mathbf{x}_t$ as a random variable : this is the Bayesian approach. The distribution of that random variable is the so called *prior distribution* ; it incorporates various prior beliefs that the system modeller has about $\mathbf{x}_t$ such as its mean, variance, tails, etc... Various distributions can be considered (Poisson, Gaussian, Pareto, mixture of distributions, ...). It is also possible to take into

account some dependence (correlation, etc...) between the different OD couples, or some dynamic model such as hidden Markov models, fractional Brownian motion, etc... although most of the time, independence between the different OD couples, as well as independence between successive traffic matrices $\mathbf{x}_t$ is supposed. When a statistical model is specified (Bayesian approach), two problems are typically considered :

- One problem is to estimate the parameters of the prior distribution (means, variances, etc...) from the constraints $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ induced by successive link values $\mathbf{y}_t$. Those parameters are typically mean values λ in the Poisson model, or means and variances in the Gaussian model, but it can be extended to more complicated models such as dynamic models in which case parameters can be transition probabilities for example. Different algorithms can be used to estimate the model parameters, each algorithm corresponds to a different criterion and results in a different estimate. Classical criteria are first and second order moments and maximum likelihood. In the case of first and second moments, the parameter are chosen so as to fit the observed means and variances/covariances between link counts. In the case of maximum likelihood estimation, the parameters are chosen so that the link values $\mathbf{y}_t, t = 1, \ldots, T$ are the most likely, that is to say that their probability density function is maximum.

- In some cases the parameters (means, variances, etc...) of the model are known quantities (or considered so). Nevertheless SNMP link measurements $\mathbf{y}_t$ provide additional information on the OD matrices $\mathbf{x}_t$ (linear constraints). These additional constraints modify the estimated values of the OD flows $\mathbf{x}_t$. If the constraints $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ are taken into account, the distribution of the OD matrix is modified. The resulting distribution is called *posterior distribution*. In the Gaussian case, the posterior distribution is still Gaussian and its mean and variance/covariance are obtained analytically (regression line). In the other cases, the posterior distribution is not a classical one, but it can nevertheless be sampled from if one uses specific algorithms, namely Markov Chain Monte Carlo (MCMC) algorithms. MCMC algorithms produce a set $\mathbf{x}_t^{(i)}, i = 1, 2, 3, \ldots$ of traffic matrices $\mathbf{x}_t$, and this set is approximately distributed as $\mathbf{x}_t$ given that $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$. Various quantities such as mean, variances or quantiles can then be calculated from that set as sample means.

Apart from Bayesian methods, non-statistical methods can be used to estimate the OD matrix. Bayesian methods produce good estimates of the OD matrix if the prior distribution is accurate. If not, this can cause serious bias. In that case, non-statistical methods such as gravity models or pseudo-inverse methods can be used to estimate the OD traffic matrix. The statistical methods based on the first and second moments only are also more robust to model misspecifications than the Bayesian ones, which are based on a prior distribution that can be inaccurate. Moreover the numerical complexity of non-statistical methods is usually lower than that of statistical methods. On the other hand, if the prior distribution is accurate, Bayesian methods produce better estimates than non-statistical ones. Therefore, non-statistical methods can be used to produce a first estimate $\hat{\mathbf{x}}_t$ of the OD traffic matrix ; then a statistical model can be calibrated on $\hat{\mathbf{x}}_t, t = 1, \ldots, T$ and Bayesian methods, with inputs the prior distribution and the SNMP measurements $\mathbf{y}_t$ can be used to improve the estimates $\hat{\mathbf{x}}_t$.

2.3 Off-line versus on-line estimation

Existing works have looked into the problem of *off-line* estimation of the OD matrix, that is to say that SNMP measurements $\mathbf{y}_t$ are stored, in a database for example, or in text files, for $t = 1, \ldots, T$ and later on, the OD matrix $\mathbf{x}_t, t = 1, \ldots, T$ is estimated. Therefore, using this information for traffic engineering assumes some form of stationarity of the OD matrix. Nevertheless, on a backbone network, the OD demands are not stationary. There are for example, some day/night effects, as well as some variations of the traffic demand over the successive days (working/non-working days for example). Nevertheless, it is true that the OD demands values are similar if one considers the same hour of the same day in two different weeks for

example, or, to a lower extent, the same hour on two consecutive days, so that some information on the present OD demand can be obtained from past SNMP measurements.

Nevertheless, if *dynamic* traffic engineering is considered the OD demands must be estimated *on-line*, that is to say that, each time new SNMP measurements $\mathbf{y}_t$ are obtained, the estimate $\hat{\mathbf{x}}_t$ of the OD matrix must be brought up to date. This means that the variations of the OD matrix on the time scale of SNMP measurements are followed *on-line*, on the basis of the SNMP information. For doing so, various estimation techniques could be used, such as the Kalman filter, the extended Kalman filter or particle filters, although the existing literature has focused on the *off-line* estimation of the OD matrix. The estimation technique essentially depends on the supposed dynamic model of the OD flows. In the case of a linear Gaussian model for example (typically a gaussian AutoRegressive Moving Average, ARMA, model), the Kalman filter must be used, since it produces the optimal estimate of the OD matrix given the past measurements (minimum mean squared error estimate). In the Gaussian case, this optimal estimate is the expected value $\hat{\mathbf{x}}_t = \mathbb{E}(\mathbf{x}_t \mid \mathbf{y}_t, \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ldots)$ given all past SNMP measurements. This is not computed from scratch each time new SNMP measurements $\mathbf{y}_t$ are obtained. On the contrary, $\hat{\mathbf{x}}_t$ is obtained from the new SNMP measurements $\mathbf{y}_t$, the previous estimate $\hat{\mathbf{x}}_{t-1}$ and joint quantities (typically filter variances,...) that are computed on-line. When the dynamic model is Gaussian but not linear, the extended Kalman filter can be used ; in that case, the model equations are linearized locally around $\hat{\mathbf{x}}_t$ and the Kalman equations are applied to the linearized model. In the general non-linear non-Gaussian case, particle filters can be used [4]. This last algorithm is based on the iterative simulation of N different trajectories $(\mathbf{x}_t^{(i)}, t = 1, 2, 3, \ldots), i = 1, \ldots, N$, so called "particles", and jointly the iterative estimation of their likelihood $p(\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_t^{(i)} \mid \mathbf{y}_1, \ldots, \mathbf{y}_t)$ given the SNMP measurements. Once more, the simulation is not started again from scratch each time new SNMP information is obtained but, on the contrary, the N particles $\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{t-1}^{(i)}), i = 1, \ldots, N$ are continued with a new sample $\mathbf{x}_t^{(i)}$ each time new SNMP measurements $\mathbf{y}_t$ are obtained. Various quantities such as an estimate of the average OD matrix, or an estimate of its quantiles, at time t, can then be obtained from these particles and their likelihood.

3 Penalization methods

As we have previously seen, the number of solutions of the system $\mathbf{y} = \mathbf{A}\mathbf{x}$ is infinite since $\mathbf{A}$ is not invertible. It is therefore necessary to select one solution (or a set of solutions) of that linear system as "better" than the other ones. To do so, different criteria can be considered, among which the minimum Euclidean norm (Section 3.1) or the second order moments (Section 3.2).

3.1 Pseudo-inverse solution

A classical method [5] in the case of underdetermined linear systems is to select the solution with minimum Euclidean norm, that is to say, to solve the following problem :

$$\begin{cases} \text{Minimize} & \|\mathbf{x}\|^2 = \sum_{i=1}^{c} (x_i)^2 \\ \text{with} & \mathbf{y} = \mathbf{A}\mathbf{x} \end{cases} \tag{2}$$

The solution of that convex optimization problem with linear constraints is $\hat{\mathbf{x}} = \mathbf{A}^*\mathbf{y}$ where $\mathbf{A}^*$ is the pseudo inverse of the routing matrix $\mathbf{A}$. Note that this solution can be applied to obtain each OD matrix $\hat{\mathbf{x}}_t = \mathbf{A}^*\mathbf{y}_t$ for each time period t, but it can also be used to obtain an average OD matrix $\lambda = \mathbf{A}^*\bar{\mathbf{y}}$ from the time averaged link values $\bar{\mathbf{y}}$, where $\bar{\mathbf{y}} = (1/T)\sum_{t=1}^{T} \mathbf{y}_t$ is the average link values vector, and $\lambda = \mathbb{E}(\mathbf{x}_t)$ is the average OD demands vector.

Unfortunately, minimizing the Euclidean norm $\|\mathbf{x}\|$ is not a good criterion in that case, since this criterion strongly penalizes the mice/elephants configurations, which are the practical case. Indeed it is common on

a backbone network that some OD demands (so called *elephants*) are 1000 times bigger than other OD demands (so called *mice*). Moreover, there is no positiveness guaranteed in the sense that, some of the OD pairs might have negative values. This possibly explains why this classical pseudo-inverse method has been left apart to estimate the OD demands matrix, and why no reference is done to that method in the OD matrix estimation literature.

3.2 A method based on the first and second order moments

The vector of average OD demands λ is not identifiable from the average link values $\mathbf{y} = \mathbf{A}\lambda$ since $\mathbf{A}$ is not invertible (too few linear relations between the average links and OD values). In order to increase the number of relations between links volume and OD demands, it is possible to take into account not only the average links volume, but also their variances and covariances. Let $\Sigma_{\mathbf{y}}$ be the variance/covariance matrix of the links volume, $\Sigma_{\mathbf{y}} = \mathbb{E}((\mathbf{y}_t - \bar{\mathbf{y}})(\mathbf{y}_t - \bar{\mathbf{y}})')$ and let equivalently $\Sigma_{\mathbf{x}} = \mathbb{E}((\mathbf{x}_t - \lambda)(\mathbf{x}_t - \lambda)')$ be the variance/covariance matrix of the OD values. It can be deduced from $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ that $\Sigma_{\mathbf{y}} = \mathbf{A}\Sigma_{\mathbf{x}}\mathbf{A}'$.

If moreover the OD demands are supposed to be independent and if some relation between the mean and variance of each OD demand is supposed, then the problem is identifiable with respect to λ, that is to say that an unique value of λ can be obtained from the means, variances and covariances values of the links. Usually a power law relationship between the mean and the variance of the OD values is supposed [2] in the form of $(\sigma_i)^2 = \phi(\lambda_i)^\alpha$, with typically $1 < \alpha < 2$ (in the Poisson case, α is equal to 1, but some statistical analysis of measured traffic have proved that in practice $1 < \alpha < 2$). With these assumptions, $\Sigma_{\mathbf{x}} = \phi\ \mathrm{diag}((\lambda_1)^\alpha, \ldots, (\lambda_c)^\alpha)$ and the mean OD values λ are the solution of the following (non-linear) system :

$$\begin{cases} \bar{\mathbf{y}} = \mathbf{A}\lambda \\ \Sigma_{\mathbf{y}} = \phi\mathbf{A}\mathrm{diag}((\lambda_1)^\alpha, \ldots, (\lambda_c)^\alpha)\mathbf{A}' \end{cases} \tag{3}$$

λ is identifiable from that system. Indeed, if N is the number of network nodes, the number of OD couples is $N(N-1)$. The number of network links is only of the order of N (say $5N$ if there are 5 links per node on the average), so that in $\bar{\mathbf{y}} = \mathbf{A}\lambda$ the number of unknowns is much higher than the number of linear constraints, and therefore λ is not identifiable from the system $\mathbf{y} = \mathbf{A}\lambda$. On the contrary, the number of link variances/covariances is $N(N+1)/2$ and moreover, the network operator knows most of the time which OD demands are always null. Therefore, λ is identifiable from the system (3), that is to say that a single value of λ can be obtained from (3), since the number of unknowns and the number of constraints are of the same order N^2 of magnitude in the System (3). As a conclusion, introducing a mean/variance relationship for the OD demands therefore makes the average OD demands identifiable from the means and the variances/covariances of the links volume.

Nevertheless, this method has a number of limitations. The first limitation is the mean variance relationship on which the method is based. This is indeed a strong assumption and there is no guarantee that this relationship is always verified. If not, the method can produce very rough results. The second limitation is the asumption of second order stationarity (stationarity of the means and variances/covariances of OD demands). Indeed, this method requires that there is a sense in speaking of means and variances/covariances of OD demands, or, in other words, that the OD demands are second order stationary. Indeed, on the timescales considered (usually several days, with SNMP measurements each 5 or 10 minutes) backbone traffic is not stationary but cyclostationary [6], that is to say that means and variance/covariances are periodic with time, with a typical period of 24 hours. And this problem cannot be solved by a local stationarity argument. Indeed, SNMP requests cannot be sent at a faster rate than approximately one every 5 minutes, which makes 12 link measurements $\mathbf{y}_t$ per hour. A minimum of 10 to 20 successive samples are necessary to estimate the average link values and their covariances, that is to say approximately one or two hours of measurements. On a timescale of several hours, second order stationarity is a doubtful assumption. A third limitation for using this method is routing instability. Indeed, in that method, the routing matrix $\mathbf{A}$ is not

time dependent. This is a major restriction since the considered timescales go from a few hours to several days of traffic and on these timescales, routing is likely to change.

In order to overcome these limitations, various improvements can be imagined, although none of these improvements have been published in the literature. For example, one solution to solve the second order non-stationarity problem would be to consider the OD values as cyclostationary signals, and use standard estimation techniques of cyclostationary signals to estimate the OD matrix. Another solution would be to consider the series of SNMP measurements at the same time (say, 2 pm) on successive days because this time series is stationary (the same time of the day is always considered), and to produce the average OD demands vector λ by solving System (3) where $\bar{\mathbf{y}}$ and $\Sigma_{\mathbf{y}}$ are the means and variances/covariances of the links volume at that time h (say, h = 2 pm). The same process can be repeated for each time h of the day so that a time dependent average OD matrix with periodicity 24 hours can be obtained. Different time granularities can be chosen, the finest granularity corresponding to the rate at which SNMP requests are sent. This method needs the SNMP and routing measurement campaign to be sufficient long (at least a few weeks) for the means and variances/covariances of the links volume to be computed with a sufficient accuracy for each time of the day.

4 Gravity models and their generalization

4.1 Gravity models

There is a long history of gravity models in telephone networks. Indeed, these models have been used for a long time to dimension circuit switched telephone networks. Gravity models can be used to estimate Internet OD volume as well. Gravity models are based on the assumption that the Origin (source) and the Destination (sink) of a traffic flow are independent. As a result, in these models, the volume of traffic flow x_{OD} with origin O and destination D is proportional to $x_{0\bullet}$ the total volume of traffic with source O, and to $x_{\bullet D}$ the total volume of traffic with destination D :

$$x_{OD} \propto x_{O\bullet} \times x_{\bullet D} \tag{4}$$

In the case of circuit switched telephone networks, the total traffic with origin O (respectively with destination D) is proportional to the population of that zone. In the case of an IP backbone network, $x_{0\bullet}$ and $x_{\bullet D}$ can be obtained easily from SNMP measurements. Gravity models are therefore easy methods to estimate Origin Destination volume on the network.

Apart from their simplicity, gravity models have a number of advantages to estimate the OD demands matrix. First of all, these models are not based on any assumption on the probability distribution of the traffic (for example, Poisson or Gaussian traffic). From that point of view no misspecification can occur. A second advantage is that the routing information is not necessary to use gravity models. This is a major advantage, since the measurements of routing information can cause a series of difficulties, among which the synchronization between SNMP and routing measurements, heterogeneous databases, routing instabilities. A possible limitation of gravity models is the Origin Destination independence assumption. This assumption is probably inaccurate to a given extent : for example, France and Quebec are French speaking areas, and therefore the volume of traffic between France and Quebec is maybe greater than what gravity models would estimate. This is true, in particular, on telephone networks but probably to a lower extent on IP networks, where *http* traffic is the majority. Gravity models are nevertheless able to produce easily rough estimates of the OD demand matrix, that can be used as starting points in more sophisticated estimation methods.

As one can see from Equation (4), the OD matrix estimate $\hat{\mathbf{x}}_t$ produced by gravity models is not a solution of the linear system $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$. Indeed, gravity models make use of the SNMP volume measured on the edge nodes, but not of the SNMP volume on the core nodes and of the routing information of Origin Destination flows in the core. If the routing information and if the SNMP information in the core are available, it is

possible to improve $\hat{\mathbf{x}}_t$ by an orthogonal projection on the set (hyperplane) of all matrices $\mathbf{x}_t$ such that $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$. Indeed, the projected value $\tilde{\mathbf{x}}_t$ of the gravity estimate $\hat{\mathbf{x}}_t$ is closer to the real value $\mathbf{x}_t$ than the gravity estimate $\hat{\mathbf{x}}_t$ as one can see from Figure 2.

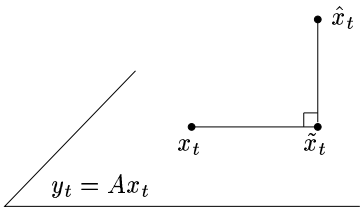


Figure 2: Projection on the hyperplane $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$.

The orthogonal projection of the gravity estimate $\hat{\mathbf{x}}_t$ on the set $\{\mathbf{x}_t : \mathbf{y}_t = \mathbf{A}\mathbf{x}_t\}$ results in practical cases in an important reduction of the root mean squared error, as we will see in Section 8 (in that case, 43% of root mean square error reduction is obtained).

4.2 Gravity models generalization

Another method based on the Origin Destination independence assumption has been proposed by Donoho et al. [7]. This method can be seen as a generalization of gravity models. Similarly to the methods of Section 3, it is also a penalization of the linear system $\mathbf{y} = \mathbf{A}\mathbf{x}$, but in this case the penalization criterion is the dependence between source and destination of traffic. To be more precise, the distribution of traffic over the different source-destination pairs is searched for, with the links volume $\mathbf{y} = \mathbf{A}\mathbf{x}$ and the source-destination independence as constraints.

Source-destination dependence can be measured as the mutual information $I(S, D)$ between source S and destination D. Let $p(s, d)$ be the probability that one traffic unit (typically one byte) has source s and destination d. Then the source-destination mutual information is equal to

$$I(S, D) = \mathbb{E}(\log(\frac{p(S, D)}{p(S)p(D)}) = \sum_{s,d} p(s, d) \log \frac{p(s, d)}{p(s)p(d)} \tag{5}$$

where $p(s) = \sum_d p(s, d)$ (respectively $p(d) = \sum_s p(s, d)$) is the probability that one traffic unit has source s (respectively destination d). The source-destination mutual information is also the Kullback-Leibler divergence between the joint source-destination distribution $p(s, d)$ and the product distribution $p(s)p(d)$. For example, when source and destination are completely independent, then $I(S, D) = 0$ since $p(s, d) = p(s)p(d)$. Minimizing the dependence between source and destination therefore comes to minimizing the source-destination mutual information $I(S, D)$.

Let now N be the total volume of traffic on the network (sum of all source-destination volume), then the average traffic volume on link l is $N \sum_{s,d} A_{(s,d;l)} p(s, d)$ where $A_{(s,d;l)}$ is the probability that one traffic unit with source s and destination d is routed through link l. Divergence from the SNMP links volume constraints $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be measured as the quadratic distance between the measured SNMP links volumes $y(l)$ and their average values $N \sum_{s,d} A_{(s,d;l)} p(s, d)$, that is to say that $\sum_l [y(l) - N \sum_{s,d} A_{(s,d;l)} p(s, d)]^2$ must be minimized with respect to $p(s, d)$.

If now both the SNMP links volume $y(l)$ constraints and the source-destination independence are balanced, the estimation of the source-destination probabilities $p(s, d)$ comes to minimizing a balanced sum of the two criteria :

$$\text{Minimize} \quad \sum_l [y(l) - N \sum_{s,d} A_{(s,d;l)} p(s, d))^2]^2 + \lambda^2 I(S, D) \tag{6}$$

where the minimization is performed with respect to $p(s,d)$, and where λ^2 is a coefficient that can be tuned to give more importance to the first or to the second criterion.

Contrary to gravity models which are based on SNMP information on the edge nodes only, this method includes SNMP information on the core nodes as well. Therefore, the results of this method are more accurate than the results of the gravity method and the gravity method can be seen as a particular case of that method. But this method needs routing informations that are not always easy to obtain, as mentioned previously.

5 Expectation Maximization (EM) algorithm

5.1 Missing data

The estimation of the OD demands matrix is a typical case of a problem with *missing data*, that is to say a problem in which some data, so called missing, are not observed and the observations are a function of the missing data (usually a random function). Typical cases of problems with missing data are mixture models (mixture of distributions) and hidden Markov models (discrete or continuous time Markov chains in noise).

In our case, the missing data are the OD volume $\mathbf{x}_t$, and the observations are the links volume $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$. Indeed, the OD matrix $\mathbf{x}_t$ is not observed directly but through the links volume $\mathbf{y}_t$. In the problems with missing data, the missing data are considered as random variables, and the observations are usually a random function of the missing data, although deterministic functions like $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$ is also possible.

Usually the distribution of the missing data is parametric (Poisson with mean λ, Gaussian with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$, etc...), but the values of the parameters θ are unknown. One problem is then to estimate the parameters θ (mean, variance, covariances, etc...) from the observed data only, as these data are the only data available to statistical analysis. Different estimation criteria can be considered, as second order moments or maximum likelihood, in which case one obtains the value of the parameters for which the observations are the most likely.

5.2 EM tutorial

Maximum likelihood estimation of parameters in the case of missing data is usually obtained by carrying out an Expectation Maximization (EM) algorithm [8]. Let $\mathbf{x}_{1:T} = (\mathbf{x}_t, t = 1, \ldots, T)$ be the missing data, and let $\mathbf{y}_{1:T} = (\mathbf{y}_t, t = 1, \ldots, T)$ be the observations. The likelihood $p(\mathbf{y}_{1:T}; \theta)$ of the observations cannot be computed, since this would require considering all possible values $\mathbf{x}_{1:T}$ of the missing data, and adding up the joint likelihood $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$ of $\mathbf{x}_{1:T}$ and $\mathbf{y}_{1:T}$ with respect to all possible values $\mathbf{x}_{1:T}$: $p(\mathbf{y}_{1:T}; \theta) = \sum_{\mathbf{x}_{1:T}} p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$. This cannot be done in practice since the number of values $\mathbf{x}_{1:T}$ is exponential with T (indeed, if each $\mathbf{x}_t$ takes K values then the number of values for $\mathbf{x}_{1:T}$ is K^T).

Rather than maximizing the observed likelihood $p(\mathbf{y}_{1:T}; \theta)$ directly, since this likelihood cannot be practically calculated, the EM algorithm is based on the maximization of the joint likelihood $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$. Indeed the joint likelihood $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$ can be computed easily or, to be more precise, with a linear complexity with respect to T. In the M1-M0 hidden Markov chain case for example, $\mathbf{y}_t$ depends on $\mathbf{x}_t$ only, and $\mathbf{x}_t$ depends on $\mathbf{x}_{t-1}$ only, so that the joint likelihood can be factorized as $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta) = p(\mathbf{x}_1; \theta) \prod_{t=2}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}; \theta) \prod_{t=1}^{T} p(\mathbf{y}_t \mid \mathbf{x}_t; \theta)$ and, as one can see, the complexity is therefore linear with T.

Therefore, if $\mathbf{x}_{1:T}$ were not missing, it would be possible to compute the joint log-likelihood $\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$ and to maximize this log-likelihood with respect to θ with different optmization techniques (Newton algorithm for example...). But, as previously mentioned, $\mathbf{x}_t$ are missing data and therefore, it is not possible to compute $\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$ and to maximize it with respect to θ, even though the complexity is linear with respect to T.

This is why an intermediate function Q is introduced. Since the data $\mathbf{x}_{1:T}$ are missing, their distribution given the observations $\mathbf{y}_{1:T}$ serves as a substitute. The function Q is put in place of the joint log-likelihood $p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$. Q is the expected value of the joint log-likelihood $\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$, where the expected value must be understood in the meaning of the distribution of the missing data $\mathbf{x}_{1:T}$ given the observations $\mathbf{y}_{1:T}$:

$$Q(\theta, \theta') = \mathbb{E}(\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta) \mid \mathbf{y}_{1:T}; \theta') \tag{7}$$

Q is a function of two variables θ and θ', where θ is the parameter of the joint log-likelihood $\log p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}; \theta)$, whereas θ' is the parameter of the distribution of $\mathbf{x}_{1:T}$ given $\mathbf{y}_{1:T}$. The EM algorithm then consists of maximizing iteratively $Q(\theta, \theta')$ with respect to the first parameter θ. More precisely, the EM algorithm is an iterative algorithm, and if θ_k is the parameter estimate produced by the previous iteration, then one additional iteration of the algorithm produces a new estimate θ_{k+1} where :

$$\theta_{k+1} = Arg \max_{\theta} Q(\theta, \theta_k) \tag{8}$$

where $Arg \max_\theta$ means that θ_{k+1} is the value of θ for which $Q(\theta, \theta_k)$ is maximum.

Each iteration of the EM algorithm is usually decomposed into two steps, the E step (E is for Expectation) and the M step (M is for Maximization). The E step consists of computing the distribution of $\mathbf{x}_{1:T}$ given $\mathbf{y}_{1:T}$ and θ_k, since that distribution is required to calculate $Q(\theta, \theta_k)$. This can be performed in different manners, depending of the statistical model considered. For example, in the case of hidden Markov models, the E step consists of a *forward backward* algorithm [9], but as previously mentioned, the exact form of the E step depends of the statistical model considered.

Once $Q(\theta, \theta_k)$ has been computed (E step) it is maximized with respect to θ : $\theta_{k+1} = Arg \max_\theta Q(\theta, \theta_k)$. This second step is the Maximization step (M step). In some cases, the maximization step is analytical, that is to say that an analytical formulation exists for θ_{k+1}. In other cases, numerical methods must be used to obtain θ_{k+1}. Once the E step and the M step have been performed, the new value θ_{k+1} takes the place of the previous estimate θ_k, and the process of iterating between E step and M step starts over again...

5.3 EM convergence and initialization

It can be proven from Equation (8) that the observations log-likelihood $\log p(\mathbf{y}_{1:T}; \theta)$ is increased by each new iteration of the algorithm, that is to say that $\log p(\mathbf{y}_{1:T}; \theta_{k+1}) \geq \log p(\mathbf{y}_{1:T}; \theta_k)$. The parameters estimate θ_k therefore converges to a maximum of the observations likelihood (or a saddle point) when the number of iterations of the algorithm increases. In practice, between 5 and 10 iterations of the algorithm are usually needed for convergence. The EM algorithm is therefore an algorithmic manner to find the maximum likelihood estimate of parameters when some data are missing.

One restriction of this algorithm is that it can converge to a *local* but not *global* likelihood maximum, if this likelihood is not convex with respect to θ. This practical case can occur in particular when real (measured) data are processed rather than synthetic ones. In that case, model misspecifications result in several local maxima of the observations likelihood $p(\mathbf{y}_{1:T}; \theta)$ and the EM algorithm will in practice converge to one of these local maxima if the starting point θ_0 is far from the global maximum θ^*.

Therefore, the EM algorithm initialization deserves a special care, especially when measured and not synthetic data are processed. To initialize the EM algorithm, it is usually convenient to produce a first estimate θ_0 by a non Bayesian method (for example, a second order moments method or a non statistical method), since non Bayesian methods are usually less sensitive to models misspecifications. Then the first iteration of the EM algorithm produces $\theta_1 = Arg \max_\theta Q(\theta, \theta_0)$ from θ_0 and the observations $\mathbf{y}_{1:T}$, a second iteration of the algorithm produces θ_2 from θ_1 and $\mathbf{y}_{1:T}$, etc... and after 7 or 8 iterations convergence to a maximum θ^* of the observations likelihood $p(\mathbf{y}_{1:T}; \theta)$ is obtained.

5.4 EM algorithm to estimate the OD demand matrix

As previously mentioned, the OD matrix estimation is a classical case of missing data problems, and the EM algorithm is a natural solution for this kind of problems. Indeed, it has been proposed to estimate the OD demand matrix, or , more precisely, its statistical parameters (OD flow means, variances, etc...) for various OD flow models. Cao et al. [2] have described the EM estimation in the Gaussian case : in that case, the mean $\mathbf{m}$ and variance Σ of the OD demands matrix is estimated. Vardi [10] has proposed the EM algorithm to estimate the average OD flow values λ in the Poisson case.

In the Gaussian case [2], the OD demands are supposed to be independent and Gaussian. Moreover, a mean-variance power law relationship is supposed, in the form of $\text{var}(x^i) = \phi(\lambda_i)^c$ where x^i is the volume of one OD couple. The distribution of $\mathbf{x}_t$ is therefore $N(\lambda, \mathbf{\Sigma})$ with $\Sigma = \phi\ \text{diag}(\lambda^c)$. The value of c is supposed to be known (a typical value is $c = 2$) and the values of λ and ϕ are estimated from the SNMP values $\mathbf{y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ by an EM algorithm. In this case, the E step of the algorithm is analytical. Indeed, the E step consists of calculating the distribution of $\mathbf{x}_t$ given $\mathbf{y}_t$ and $\theta_k = (\lambda^{(k)}, \Sigma^{(k)})$ and this distribution is Gaussian, with mean $m_t^{(k)}$ and covariance matrix $R^{(k)}$:

$$\begin{aligned} m_t^{(k)} &= \mathbb{E}(\mathbf{x}_t \mid \mathbf{y}_t; \theta^{(k)}) = \lambda^{(k)} + \Sigma^{(k)} A'(A\Sigma^{(k)}A')^{-1}(\mathbf{y}_t - A\lambda^{(k)}) \\ R^{(k)} &= var(\mathbf{x}_t \mid \mathbf{y}_t; \theta^{(k)}) = \Sigma^{(k)} - \Sigma^{(k)} A'(A\Sigma^{(k)}A')^{-1} A\Sigma^{(k)} \end{aligned} \tag{9}$$

where $\lambda^{(k)}$ and $\Sigma^{(k)}$ are the estimates of λ and Σ produced by the previous iteration of the EM.

As in this case the successive values $\mathbf{x}_t, t = 1, \ldots, T$ of the OD matrix are independent, and as $\mathbf{y}_t$ is a deterministic function of $\mathbf{x}_t$ since $\mathbf{y}_t = \mathbf{A}\mathbf{x}_t$, the function $Q(\theta, \theta_k)$ of Equation (7) is also equal to $Q(\theta, \theta_k) = \sum_{t=1}^{T} \mathbb{E}(\log p(\mathbf{x}_t, \theta) \mid \mathbf{y}_t; \theta_k)$. If now the Gaussian log-likelihood is put in place of $\log p(\mathbf{x}_t; \theta)$ and if the expectation $\mathbb{E}(\bullet \mid \mathbf{y}_t; \theta^{(k)}$ is taken in the meaning of $N(m_t^{(k)}, R^{(k)})$ (distribution of $\mathbf{x}_t$ given $\mathbf{y}_t$ and $\theta^{(k)}$), the function $Q(\theta, \theta^{(k)})$ can now be writen as :

$$Q(\theta, \theta^{(k)}) = -\frac{T}{2}[\log|\mathbf{\Sigma}| + tr(\mathbf{\Sigma}^{-1} R^{(k)})] - \frac{1}{2}\sum_{t=1}^{T}(m_t^{(k)} - \lambda)'\mathbf{\Sigma}^{-1}(m_t^{(k)} - \lambda) \tag{10}$$

The maximization step (M step) of the algorithm then comes to maximizing (10) with respect to λ and ϕ with $\mathbf{\Sigma} = \phi \text{diag}(\lambda^c)$, so as to produce $\lambda^{(k+1)}$ and $\mathbf{\Sigma}^{(k+1)}$. This maximization is performed by a numerical method since there is no analytical solution to this maximization problem.

The EM algorithm in the case of a Gaussian model with a mean-variance power-law relationship has been proposed and validated by Cao et al. [2] on some traffic measured on a router in a Local Area Network of Lucent. As the stationary Gaussian assumption is not true on this dataset, Cao et al. have used a local stationarity argument : the above described EM algorithm is carried out on a sliding window, thus producing a time varying estimate of (λ, ϕ). The major limitation of this approach is the fact that the EM algorithm must be carried out for each window, and that the computational load is therefore very heavy, since the EM algorithm is already an iterative algorithm and since moreover, in each iteration of this algorithm, the maximization step must be performed numerically.

In some previous work, Vardi [10] had described the EM estimation of the mean λ of the OD demands matrix in the Poisson case. We will not describe in all its details the EM algorithm to estimate the OD demands matrix from the routing and SNMP information, in the case of Poisson traffic, since this is very similar to the Gaussian case. It is nevertheless important to remark that in the Poisson case, the E step is not analytical, that is to say that the distribution of $\mathbf{x}_t$ given $\mathbf{y}_t$ and $\lambda^{(k)}$ is not easily obtained. To overcome that problem, Vardi approximates, in the E step of the EM algoritm, the Poisson distribution with mean λ by a Gaussian distribution with mean $\mathbf{m} = \lambda$ and with covariance matrix $\Sigma = \text{diag}(\lambda)$, so as to make the E step analytical. The rationale for doing so is the fact that, when the average OD demands λ have big values the Gaussian distribution is a valid approximation for the Poisson distribution.

6 Markov Chain Monte Carlo (MCMC) algorithms

As mentioned in Section 2, a possible approach to infer the OD matrix $\mathbf{x}$ from the links volume $\mathbf{y}$ and the routing matrix $\mathbf{A}$ is the Bayesian one [11]. Tebaldi et al. [11] use Markov Chain Monte Carlo (MCMC) algorithms to infer $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{A}$. In this section, we describe and discuss MCMC algorithms and, in particular, the Gibbs algorithm and the Hastings Metropolis algorithm, into all their details, and then we discuss their possible use to estimate the OD demands matrix (Tebaldi, [11]).

6.1 MCMC tutorial

In general, MCMC algorithms are intensive computer simulation algorithms that can be used to solve a variety of problems, one of these problems being the inference of unobserved data (so called *hidden* or *missing* data or *latent* data) in the case when some *observations* are at disposal and these observations are deterministic or random functions of the hidden data. Hidden data models are also the framework in which EM algorithm can be used. The difference between the EM algorithm and MCMC algorithms is that the EM algorithm estimates the OD flows statistical parameters (mean, variance, etc...), whereas MCMC algorithms produce, by simulation, a set of traffic matrices $\mathbf{x}$ that are approximately distributed as $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{A}$. Another major difference is that EM estimation of the traffic matrix parameters has been studied in the Gaussian and Poisson cases, and other models would need additional work (the exact form of the E and M steps indeed differs from one model to another), whereas MCMC algorithms can be used for any OD flows distribution, without any additional work, provided that the probability density function (pdf) of the OD flows can be calculated.

The principle of any MCMC algorithm (Gibbs algorithm, Hastings Metropolis algorithm, ...) is to generate, by computer simulation, a discrete time Markov chain that converges in distribution to a given *target* distribution (specified by the software developer). By doing so, one obtains a series of samples that are approximately distributed as the target distribution, if one lets the MCMC algorithm be carried out a sufficiently long time and if the first samples produced by the algorithm are left apart (*warm up* period).

The target distribution is specified by the software developer and it can be changed at will as this is the case for any parameter of a function (for example in *C* programming, pointer to functions can be used). In the case of hidden data, this target distribution is the so called *posterior* distribution, that is to say the distribution of the hidden data $\mathbf{x}$ given the observed data $\mathbf{y}$. This posterior distribution can be characterized by its probability density function (pdf) $p(\mathbf{x} \mid \mathbf{y})$, that is equal to the following ratio :

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{p(\mathbf{x})\, p(\mathbf{y} \mid \mathbf{x})}{\int_{\mathbf{x}'} p(\mathbf{x}')p(\mathbf{y} \mid \mathbf{x}')} \tag{11}$$

In Equation (11), $p(\mathbf{x})$ is the pdf of the *prior* distribution, that is to say the distribution of $\mathbf{x}$, when no additional information is provided by $\mathbf{y}$. This prior distribution is the assumed distribution of the hidden data $\mathbf{x}$ when no additional information is provided by some observations $\mathbf{y}$. The prior distribution $p(\mathbf{x})$ expresses the software developer's prior beliefs about the range of possible values for $\mathbf{x}$ and the likelihood of these different values. It can be based on some prior knowledge of the studied system (previous measurements, etc...) if such knowledge is available, and if not it can be more or less arbitrary (we will discuss that point later). In Equation (11), $p(\mathbf{y} \mid \mathbf{x})$ is the probability density function of $\mathbf{y}$ given $\mathbf{x}$, since the observations $\mathbf{y}$ are, in general, a random function of the hidden data $\mathbf{x}$. However, in the particular case of trafic matrix inference, the observations $\mathbf{y}$ (link counts) are a *deterministic* function of the missing data $\mathbf{x}$ (OD counts) in the form of $\mathbf{y} = \mathbf{A}\mathbf{x}$ so that in that case $p(\mathbf{y} \mid \mathbf{x})$ is a Dirac.

Most of the times, the integral $\int_{\mathbf{x}'} p(\mathbf{x}')p(\mathbf{y} \mid \mathbf{x}')$ in Equation (11) cannot be computed, neither analytically nor by numerical methods, since it is a non-classical integral over a multidimensional set (typically

$\mathbb{R}^N$). Therefore the posterior distribution is specified up to some multiplicative factor, and this factor is an unknown. In practice, one can only write :

$$p(\mathbf{x} \mid \mathbf{y}) \propto p(\mathbf{x})\, p(\mathbf{y} \mid \mathbf{x}) \tag{12}$$

where the $\propto$ sign means proportionality. MCMC algorithms are able to produce a Markov chain that converge in distribution to the posterior distribution $p(\mathbf{x} \mid \mathbf{y})$ eventhough the pdf of the posterior distribution is defined up to a multiplicative factor only. This is one of the very positive points of MCMC algorithms. Another positive point is that the prior distribution $p(\mathbf{x})$ can be changed at will, so that different models for the hidden data (say Poisson, Gaussian, Pareto, ... with or without dependence between the different components of $\mathbf{x}$, etc...) can be tested by the software developer. Doing so, it is possible to adapt to different situations, corresponding to a different *a priori* knowledge of the studied system, or to compare these different situations with each other.

As we have seen before, MCMC algorithms produce, by intensive computer simulations, a series of samples $\mathbf{x}^n, n = 1, 2, \ldots$ that are approximately distributed as $p(\mathbf{x} \mid \mathbf{y})$. This means that a histogram of this series would fit well with the pdf of the hidden data given the observations $p(\mathbf{x} \mid \mathbf{y})$. This histogram, or equivalently the series $\mathbf{x}^n, n = 1, 2, \ldots$ itself, sums up all the information concerning the hidden data $\mathbf{x}$ that one is able to get from the observations $\mathbf{y}$, and not only partial information such as the mean or the variance. Nevertheless various information can be obtained from that histogram, for example mean, variance but also covariance or quantile like quantities. For example the probability that one component x_i value is bigger than a specific threshold, or the probability that, jointly, two components x_i and x_j are over two specific thresholds, etc... can be estimated from the series $\mathbf{x}^n, n = 1, 2, \ldots$. When $\mathbf{x}$ is the OD matrix, quantiles for example could be very useful for capacity planning or load balancing since from them, one can compute a "worst case" (biggest OD pair value, biggest OD pair couple values, etc...) with a given fiability.

Derivate quantities such as mean, variance, quantiles are usually computed from the series $\mathbf{x}^n, n = 1, 2, \ldots$ as sample averages. For example the mean is approximated as $\hat{\mathbf{m}} = (1/T) \sum_{n=1}^{T} \mathbf{x}^n$, the variance/covariance matrix as $\hat{\Sigma} = (1/T) \sum_{n=1}^{T} (\mathbf{x}^n - \hat{\mathbf{m}})'(\mathbf{x}^n - \hat{\mathbf{m}})$ where $\bullet'$ denotes transposition, and the quantiles of component x_i as $\hat{G}_i(x) = (1/T) \sum_{n=1}^{T} \mathbb{1}_{x_i^n \geq x}$ where $\mathbb{1}_{x_i^n \geq x}$ is 1 if the value of x_i^n is greater than x and 0 elsewhere (counts of the number of samples such that $x_i^n \geq x$), etc...

MCMC algorithms is a family of algorithms based on the same principle, the simulation of a discrete time Markov chain that converges, in distribution, to a target distribution (for example the posterior distribution $p(\mathbf{x} \mid \mathbf{y})$ in the case of hidden data. Different algorithms belong to that family, among which the Metropolis algorithm and the Gibbs algorithm. Both algorithms are used by Tebaldi et al. [11] to estimate the OD demands matrix from SNMP measurements. This is the reason why we describe the Gibbs algorithm (Section 6.1.1) and the Metropolis algorithm (Section 6.1.2). The combination of the Gibbs algorithm and the Metropolis algorithm (so called Metropolis within Gibbs algorithm) to estimate the OD demands matrix is then described in Section 6.2.

6.1.1 Gibbs algorithm

In a general setting, the Gibbs algorithm is used to simulate a multidimensional random variable $\mathbf{z} = (z_1, \ldots, z_N)$ under its joint distribution $p(\mathbf{z})$. Usually it is impossible to sample directly from the *joint* distribution $p(\mathbf{z})$. But on the contrary, in many applications, the *conditional* distributions $p_i(z_i \mid \mathbf{z}_{-i})$ are easy to simulate, where $p_i(z_i \mid \mathbf{z}_{-i})$ is the conditional distribution of z_i when the other components (vector $\mathbf{z}_{-i}$) have fixed values.

The principle of the Gibbs algorithm is to simulate one component z_i at a time. Each component z_i is simulated under its posterior distribution $p_i(z_i \mid \mathbf{z}_{-i})$ where $\mathbf{z}_{-i}$ takes current values. The different

components of $\mathbf{z}$ are swept iteratively as follows :

$$\begin{array}{lcl} z_1^{n+1} & \leftarrow & p_1(z_1 \mid z_2^n, \ldots, z_N^n) \\ z_2^{n+1} & \leftarrow & p_2(z_2 \mid z_1^{n+1}, z_3^n, \ldots, z_N^n) \\ \vdots & & \vdots \\ z_{N-1}^{n+1} & \leftarrow & p_{N-1}(z_{N-1} \mid z_1^{n+1}, \ldots, z_{N-2}^{n+1}, z_N^n) \\ z_N^{n+1} & \leftarrow & p_N(z_N \mid z_1^{n+1}, \ldots, z_{N-1}^{n+1}) \end{array} \tag{13}$$

Equation (13) is a description of iteration $n+1$ of the Gibbs algorithm. The upper indices n (respectively $n+1$) refer to the quantities produced after n (respectively $n+1$) iterations of the algorithm.

When n is large then $\mathbf{z}^n = (z_1^n, z_2^n, \ldots, z_N^n)$ is approximately distributed under the joint distribution $p(\mathbf{z})$. The Gibbs algorithm is therefore a very simple yet powerful algorithm to simulate multidimensional random variables.

6.1.2 Metropolis algorithm

The Metropolis algorithm makes the simulation of a random variable when its probability density function is defined up to a multiplicative factor, as it is the case in Equation (12) for example. The principle of the Metropolis algorithm is to draw a random variable with a distribution that one can simulate, and to accept that random variable (so called *candidate*) with a probability that is equal to the Metropolis ratio. This ratio is a function of the likelihoods of the draw and the previous sample under the distribution that one wants to simulate (the *target* distribution) and under the distribution that one can simulate (the *instrumental* distribution). The sequence of random variables that is produced by the Metropolis algorithm converges to the *target* distribution when the number of iterations of the algorithm is large.

Let $p(\mathbf{z})$ be the pdf of the target distribution (the one from which one would like to obtain samples), and let $q(\mathbf{z})$ be the pdf of the instrumental distribution (the one from which one is able to produce samples easily). The Metropolis algorithm is an iterative algorithm, and if $\mathbf{z}^k$ is the sample produced by the previous iteration of the Metropolis algorithm, one new iteration can be described as follows :

1. Draw $\tilde{z}^{k+1}$ with distribution q(z). $\tilde{z}^{k+1}$ is the candidate.
2. Accept $\tilde{z}^{k+1}$ with probability

$$\alpha(z^k, \tilde{z}^{k+1}) = \min\{1, \frac{p(\tilde{z}^{k+1})}{p(z^k)} \frac{q(z^k)}{q(\tilde{z}^{k+1})})\} \quad \text{Metropolis ratio} \tag{14}$$

3. If $\tilde{z}^{k+1}$ is accepted then $z^{k+1} = \tilde{z}^{k+1}$; if not, then $z^{k+1} = z^k$.

z^k is the sample produced by the previous iteration of the Metropolis algorithm, and z^{k+1} is the sample produced by the new iteration. z^{k+1} is equal to the candidate $\tilde{z}^{k+1}$ if the candidate has been accepted, and if not, it is equal to the output z^k of the previous iteration. When the number k of iterations is large, then z^k is distributed as the target distribution $p(z)$.

The Metropolis algorithm is therefore an algorithm to produce samples distributed as a *target* distribution that cannot be sampled from directly, provided that the pdf $p(z)$ of this distribution is specified (up to a multiplicative factor). It is based on a recycling principle, the recycling process being in that case the random acceptance or the rejection of the *candidates*. The candidates are produced by the sampler with an *instrumental* distribution from which values are drawn easily. As the pdf $p(z)$ of the target distribution is used in the Metropolis ratio only, and as it appears both in the upper and lower parts of that ratio, it is possible to use the Metropolis algorithm when $p(z)$ is defined up to a multiplicative factor only.

6.2 Metropolis within Gibbs algorithm to estimate the OD traffic matrix

This section is a detailed description of the MCMC approach for computing the OD traffic matrix [11]. The goal is to simulate $\mathbf{x}$ under its posterior distribution $p(\mathbf{x} \mid \mathbf{y})$ with the constraint that $\mathbf{y} = \mathbf{A}\mathbf{x}$ (Equation 1). The inputs are the vector of link counts $\mathbf{y}$, the routing matrix $\mathbf{A}$, as well as a prior distribution on the OD pairs that are supposed to be independent :

$$p(\mathbf{x}) = \prod_{i=1}^{c} p_i(x_i) \tag{15}$$

In this section we follow Tebaldi et al. [11]. The routing matrix $\mathbf{A}$ is full line rank r. Then, up to some linear combinations of the lines of $\mathbf{A}$ and to some permutations of the columns of $\mathbf{A}$ one can write :

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A_1} & \mathbf{A_2} \end{array} \right] \tag{16}$$

where $\mathbf{A_1}$ is an invertible $r \times r$ matrix and where $\mathbf{A_2}$ is a $r \times (c-r)$ matrix. Naturally the same linear combinations should be applied on the components of $\mathbf{y}$ and a reordering of the OD pairs should also be performed so that Equation (1) is still true. $\mathbf{x}$ is similarly split up into an upper part of size r and a lower part of size $c-r$: $\mathbf{x} = (\mathbf{x_1}', \mathbf{x_2}')'$. Then it results from Equations (1) and (2) that:

$$\mathbf{x_1} = \mathbf{A_1^{-1}} \left(\mathbf{y} - \mathbf{A_2}\, \mathbf{x_2} \right) \tag{17}$$

where $\mathbf{A_1^{-1}}$ is the inverse matrix of $\mathbf{A_1}$.

Therefore, $\mathbf{x_2}$ is a set of free variables and the simulation of $p(\mathbf{x} \mid \mathbf{y})$ reduces to simulating $\mathbf{x_2}$ under the posterior distribution $p(\mathbf{x_2} \mid \mathbf{y})$, and then getting $\mathbf{x_1}$ from Equation (4). In [11], Tebaldi et al.suggest using a Metropolis within Gibbs algorithm to simulate $\mathbf{x}_2$ given $\mathbf{y}$.

The simulation of $\mathbf{x_2}$ under the posterior distribution $p(\mathbf{x_2} \mid \mathbf{y})$ can be performed by running a Gibbs algorithm. As the principle of the Gibbs algorithm is to update one component $\mathbf{x}_2^i$ at a time each component $\mathbf{x}_2^i$ is drawn from the distribution $p(\mathbf{x}_2^i \mid \mathbf{x}_2^{-i}, \mathbf{y})$ of that component given all the other components $\mathbf{x}_2^{-i} = (x_2^1, \ldots, x_2^{i-1}, x_2^{i+1}, \ldots, x_2^{c-r})$ and $\mathbf{y}$. It is therefore necessary to compute the probability density functions $p(x_2^i \mid \mathbf{y}, \mathbf{x}_2^{-i})$. After a few straightforward computations it is proved that :

$$p(x_2^i \mid \mathbf{y}, \mathbf{x_2^{-i}}) \quad \propto \quad p(x_2^i) \prod_{j=1}^{r} p(x_1^j) \tag{18}$$

where $\propto$ means proportionality. Equation 18 is taken directly from Tebaldi's paper ([11], eq.6), see also [12] for a demonstration. Each iteration of the Gibbs algorithm amounts to sequentially update the different components x_2^i of vector $\mathbf{x_2}$, and then to calculate $\mathbf{x_1}$ from Equation (17).

As the posterior distribution $p(x_2^i \mid \mathbf{y}, \mathbf{x_2^{-i}})$ is not a classical distribution (Poisson, Gauss, etc...) the samples $x_2^{i,(n+1)}$ can not be obtained directly. As one can see from Equation (18) the probability density function $p(x_2^i \mid \mathbf{y}, \mathbf{x_2^{-i}})$ is defined up to a multiplicative factor. Therefore, it is necessary to use a Metropolis algorithm to draw each component x_2^i.

The *target* distribution for the Metropolis algorithm is the distribution of x_2^i conditionally to $\mathbf{y}$ and to $\mathbf{x_2^{-i}}$, which probability density function is :

$$p(x_2^i \mid x_2^{-i}, \mathbf{y}) \propto p(x_2^i) \prod_{j=1}^{r} p(x_1^j) \tag{19}$$

In the equation (19), the quantities x_1^j implicitely depend on x_2^i since $\mathbf{x_1} = \mathbf{A_1^{-1}} \left(\mathbf{y} - \mathbf{A_2}\, \mathbf{x_2} \right)$ where $x_2^j, j \neq i$ and $\mathbf{y}$ have fixed values and x_2^i varies.

As described in Section 6.1.2, candidate samples are drawn from an *instrumental* distribution and these samples are randomly accepted or rejected, the acceptation ratio being the Metropolis ratio. The *instrumental* distribution can be, for example, the prior distribution $p(x_2^i)$ of x_2^i. Indeed, with this choice of prior distribution, the expression of the Metropolis ratio is simpler than in Section 6.1.2 :

$$\alpha(x_2^i, \tilde{x}_2^i) = \min\{1, \frac{\prod_{j=1}^{r} p(\tilde{x}_1^j)}{\prod_{j=1}^{r} p(x_1^j)}\} \tag{20}$$

where $\mathbf{x_1} = \mathbf{A_1^{-1}} (\mathbf{y} - \mathbf{A_2}\, \mathbf{x_2})$ and where $\tilde{\mathbf{x}}_\mathbf{1} = \mathbf{A_1^{-1}} (\mathbf{y} - \mathbf{A_2}\, \tilde{\mathbf{x}}_\mathbf{2})$.

6.2.1 Pros and cons of Metropolis within Gibbs OD demands matrix estimation

As a conclusion, it is possible to use a *Metropolis within Gibbs* algorithm to produce a set of traffic matrices $\mathbf{x}^{(n)}, n = 1, 2, \ldots$ given the vector of SNMP measurements $\mathbf{y}$. For that, it is necessary to specify the prior distribution $p(\mathbf{x})$ of the OD traffic matrix $\mathbf{x}$, that is to say the distribution of the OD traffic matrix when no information is provided by SNMP measurements. This prior distribution can be a classical traffic model (for example Poisson or Gaussian), but it can also be a traffic model of any type, provided that its probability density function is specified. The *traffic model flexibility* is one positive point of that method, together with the ability to produce *a set* of OD demands matrices and not only one single estimate, since various quantities such as quantiles can be calculated from this set. On the other hand, the computational load to carry out a *Metropolis within Gibbs* algorithm is very high, since an iterative algorithm (Metropolis) is carried out a high number of times within each iteration of another iterative algorithm (Gibbs). Computational load is the main con of this method, and its practical application to backbone networks of 50 nodes or more is probably doubtful.

7 Our method to estimate the OD traffic matrix

In Sections 3, 4, 5 and 6 various existing OD demands matrix estimation techniques have been presented. In particular, pros and cons of the different estimation techniques have been discussed. In the present section, we present the algorithm that we have developed.

This algorithm is an improvement over the Metropolis within Gibbs algorithm presented in Section 6. The improvement relies on the fact that with our method, it is possible to estimate the *prior distribution* $p\mathbf{x}_t$) of the OD flows $\mathbf{x}_t$ from the SNMP measurements $\mathbf{y}_t$, whereas in the classical Metropolis within Gibbs algorithm this prior distribution is arbitrary. In addition, a dynamical model is explicitely taken into account for each OD flow. Taking into account the dependence between the successive values taken by the same OD flow results in an important reduction of the OD matrix estimation error as this will be demonstrated in Section 8.

7.1 Importance of the prior distribution

First of all, we will show that in Bayesian estimation it is fundamental to base the estimation on a sufficiently accurate prior distribution. Let us come back to the toy example network of Figure (1) to develop an intuition of this fact. Suppose that $y_1 = x_1 + x_3 = 3$ and $y_2 = x_1 + x_2 = 5$ and let us compare the likelihood of the different solutions $(x_1, x_2, x_3) = (0, 5, 3), (1, 4, 2), (3, 2, 0)$. Two prior distributions are considered, the independent Poisson distribution with means $\lambda = (2, 2, 2)$, and the independent Poisson distribution with means $\lambda = (1, 3, 2)$. The likelihoods of the different solutions are summed up in Table 1.

As one can see from Table 1, if for example x_1, x_2 and x_3 are Poisson with means $\lambda_1 = \lambda_2 = \lambda_3 = 2$ (left part of Table 1) then the most likely solution is $(x_1, x_2, x_3) = (2, 3, 1)$ while if $\lambda_1 = 1$, $\lambda_2 = 3$ and $\lambda_3 = 2$ (lower part of Table 1) the most likely solution is this time $(x_1, x_2, x_3) = (1, 4, 2)$.

Poisson with means $\lambda = (2, 2, 2)$		Poisson with means $\lambda = (1, 3, 2)$	
$\mathbf{x} = (x_1, x_2, x_3)$	likelihood $p(\mathbf{x}; \lambda)$	$\mathbf{x} = (x_1, x_2, x_3)$	likelihood $p(\mathbf{x}; \lambda)$
(0,5,3)	0.0009	(0,5,3)	0.0067
(1,4,2)	0.0066	**(1,4,2)**	**0.0167**
(2,3,1)	**0.0132**	(2,3,1)	0.0112
(3,2,0)	0.0066	(3,2,0)	0.0019

Table 1: Likelihood of the 4 solution traffic matrices for the network of Fig. 1 with link values $y_1 = 3$ and $y_2 = 5$. Left table : $x_1 \sim P(2)$,$x_2 \sim P(2)$ and $x_3 \sim P(2)$. Right table : $x_1 \sim P(1)$,$x_2 \sim P(3)$ and $x_3 \sim P(2)$.

Moreover when $\lambda_1 = \lambda_2 = \lambda_3 = 2$ the most likely solution is $(x_1, x_2, x_3) = (2, 3, 1)$ with a likelihood value $p(\mathbf{x}; \lambda) = 0.0132$ and this solution has two direct competitors $(x_1, x_2, x_3) = (1, 4, 2)$ and $(x_1, x_2, x_3) = (3, 2, 0)$ with same likelihood values $p(\mathbf{x}; \lambda) = 0.0066$, that is to say two times less likely than $(x_1, x_2, x_3) = (2, 3, 1)$. When $\lambda_1 = 1$, $\lambda_2 = 3$ and $\lambda_3 = 2$ the most likely solution is $(x_1, x_2, x_3) = (1, 4, 2)$ with likelihood value $p(\mathbf{x}; \lambda) = 0.0167$ and this solution has one direct competitor $(x_1, x_2, x_3) = (2, 3, 1)$ with likelihood value $p(\mathbf{x}; \lambda) = 0.0112$ that is to say 33% less likely than $(x_1, x_2, x_3) = (1, 4, 2)$.

As one can see from this toy example, the prior distribution has a strong influence on the likelihood of the different solutions. Two different prior distributions therefore induce a different selection of one solution $\hat{\mathbf{x}}$ of $\mathbf{y} = \mathbf{A}\mathbf{x}$ as the most likely one. The prior distribution should therefore not be an arbitrary choice but, on the contrary, it should be an accurate picture of the real distribution. By real distribution, we mean the distribution that one would observe if it was possible to measure the OD demands directly.

But obviously, as direct measurement of the OD values is not possible, designing an accurate prior distribution is a challenging task. This is a fundamental problem in OD matrix estimation by bayesian methods that has not been sufficiently studied in the literature. Solving that problem is absolutely necessary to apply these methods to real traffic. In this Section, we develop a new estimation method that one can see as an improvement over the MCMC method of Tebaldi [11]. Our method is original in that :

1. the prior distribution $p(\mathbf{x})$ is estimated from the link measurements $\mathbf{y}_t, t = 1, 2, 3, \ldots, T$ rather than being an arbitrary choice, and

2. a dynamic model (hidden Markov model) can be introduced explicitely to take into account dependence between successive OD matrix values $\mathbf{x}_t$, rather than considering time independence.

7.2 A *divide and conquer* method

The principle of our method is to deal with the influence of both dimensions (time t, and OD pair number x_i) by a *divide and conquer* method. We divide the whole problem into two subproblems which are each treated independently (one for time dependency and one for OD pair dependency). A global convergence is nevertheless obtained by exchanging some information between the two subproblems. By doing so, we manage to keep the numerical complexity of solving the global complex problem to a reasonable degree.

Our algorithm consists of a loop (feedback) as it is displayed on Figure 2. This loop can be divided into two boxes. The first box (*Metropolis within Gibbs*) deals with the OD pair dependency. The second box (*EM algorithm*) deals with the time dependency. The process of exchanging information between the two boxes is iterated until the convergence to a fixed point is obtained. We will discuss about some issues of the loop process in the section 7.4.

The first box runs MCMC methods proposed by Tebaldi [11] to simulate the traffic matrix. Its inputs are the link counts and a prior distribution for each time measurement period. It produces as an ouput the estimated traffic matrix for each time measurement period. We will not go further in the description of this

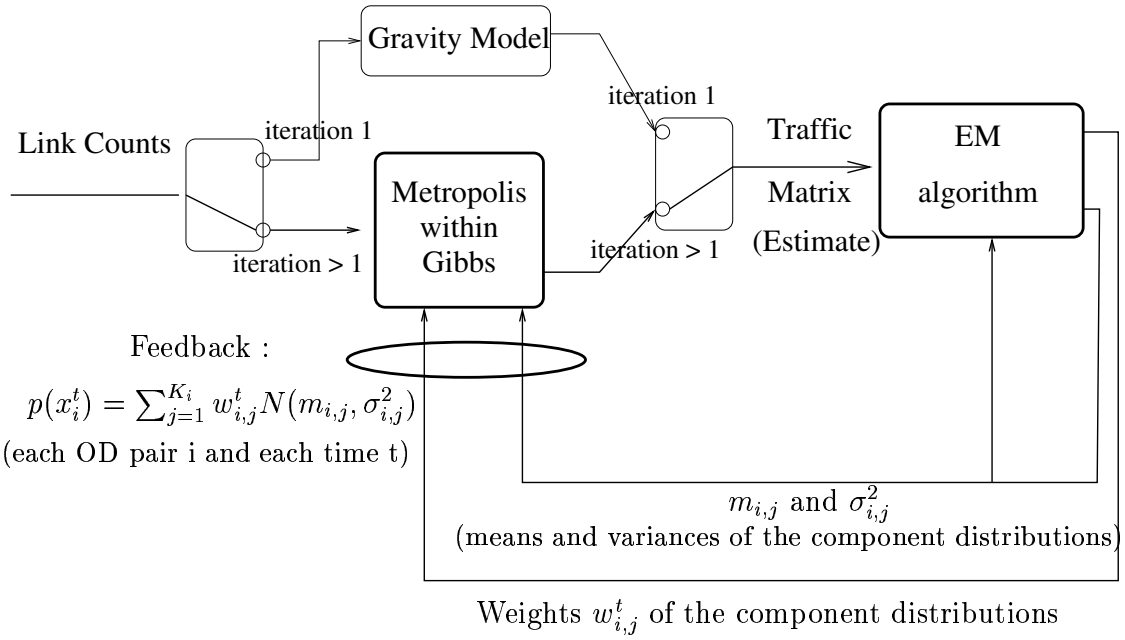


Figure 3: The global loop.

box since it relies on the Hastings Metropolis within Gibbs algorithm which has already been presented in section 6.1.2.

The second box consists of an Expectation Maximization algorithm with a BIC criterion. It fits a mixture of Gaussian distributions with an unknown number of components on the successive values taken by each OD pair. Its inputs are an estimate for the successive values of each OD pair. It produces the parameters (weights, means and standard deviations) of the mixture that best fit that OD pair for each time period. The parameters of the priors are then provided as an input to the first box (*Metropolis within Gibbs*) in a feedback loop (in the form of a new prior distribution for each OD pair and each time measurement period).

Note that the weights of the Gaussians of the mixture change over time. This means that the supposed distribution of each OD pair varies throughout the successive measurement periods. By letting the weights vary over time non stationarities in the traffic [2][13] can be taken into account. We will explain some details about the second box in section 7.3.

The algorithm has already been validated on simulated traffic in some previous papers [14][12]. In this paper the algorithm is validated on real traffic data on a single router network for which direct measurements of the OD counts were made available through specific software [13][2]. This is the purpose of section 8.

7.3 Estimating the prior distribution

In this section, we detail the second box which deals with the subproblem of time dependency. Our goal is to propose a suitable model for the *prior* distribution of each OD pair at each time measurement.

As one can see from various applications, many distributions on real data can be modelled successfully by a mixture model [15]. Moreover, the mixture model makes it possible to adapt to different distributions at each time measurement period by playing on the weights of the components in the mixture, since these weights can change at each time period. Mixtures therefore provide a flexible, general and yet very simple framework for taking into account the traffic variability.

Let consider the *successive* values taken by only *one* OD pair in time. The successive values $x_1, x_2, \ldots, x_T$

of that OD pair are distributed as a mixture of K Gaussians :

$$p(x_k) = \sum_{j=1}^{K} w_j(k) G_{\mu_j,\sigma_j}(x_k) \tag{21}$$

In Equation (7) $G_{\mu,\sigma}$ is the probability density function of the Gaussian distribution with mean μ and standard deviation σ. $w_j(k)$ is the probability that x_k is a sample from the component j of the mixture.

The problem now is to estimate the number K of components, as well as their means μ_j, standard deviations σ_j and the weights $w_j(k)$ from the successive values taken by that OD pair.

Of course the real values of the OD pair are unknown. Only estimates of these values are available. Therefore, in our global iterative algorithm we do not use the real values of that OD pair to compute the parameters of the mixture, but their estimates. At the first iteration of the algorithm, the estimates are provided by, for example, a generalized gravity model [16], then for the next iterations the estimates are these provided by the MCMC method described in Section 6.1.2.

Let us consider that K has a fixed value and we want to determine the parameters of the model which best fit the distribution of one OD pair. To be more precise, we adjust the parameters (weights, means and standard deviations) in order to maximize the likelihood of the estimated OD pair $\mathbf{x} = (x_1, x_2, \ldots, x_T)$. An Expectation Maximization (EM) algorithm [17] is used for this. We have described the principles of the EM algorithm in Section 5. As a consequence, we will not insist on this point. The main difference here is that we use the estimates of the OD pair traffic as the observations and the hidden parameters are the weights of the Gaussian components at each time.

Various approaches can be used (reversible jumps MCMC, BIC, etc...) to evaluate K. In order to keep a low computational cost we used the BIC criterion. The principle is to try different values for K (from $K = 1$ to $K = K_{\max}$) and to select the one which maximizes the BIC criterion :

$$BIC = 2 * L(K) - \nu(K) \log(T) \tag{22}$$

In Equation (11) $L(K) = \log p(\mathbf{x}; \hat{\theta}_K)$ is the log-likelihood of the mixture with K components which best fits the OD pair ($\hat{\theta}_K$ is given by the EM algorithm), and $\nu(K)$ is the number of free parameters of the model. In the case of a mixture of Gaussians $\nu(K) = 3\,K - 1$. T is the number of available measurements.

The choice of $K_{\max}$ is quite partial but some experiments in that field have proved that $T^{0.3}$ seems to be sufficient where T is the number of time periods. For example, the program MIXMOD uses this value and is considered as a reference in the field.

In the case of the mixture of Gaussians, the EM algorithm is very simple and extremely fast. The computational load that is added to Tebaldi's method by the EM algorithm is about 10 percent of the total load, if one considers the overall cost of estimating K, the μ_j, σ_j and $w_j(k)$.

7.4 The global loop

In this section, we detail the mechanism of feedback between the two boxes of our algorithm. In the first box only *one* time period is considered and we simulate the *various* OD pairs for that time period by MCMC methods. In the second box only *one* OD pair is considered and a mixture model is fitted to the *successive* values taken by that OD pair. In this section the full problem is dealt with in the sense that we estimate the *successive* values of the *various* OD pairs jointly, but with a reasonable numerical complexity (1 hour of computation on a Pentium IV 2GHz for analyzing 5 days of traffic when the time period is 5 minutes).

7.4.1 Initialization of the global loop

We want our method to be as deterministic as possible, in the sense that we do not wish to fix *anything* arbitrarily. In particular, the prior distributions of the OD pairs during the first iteration of our global loop

must not be arbitrary. In order to launch our iterative process (global loop), we would need either a first estimate of the traffic matrix or a prior distribution for the OD pairs. Unfortunately these quantities are not available directly.

The solution that we have found is to produce an initial estimate of the successive values of the traffic matrix by a very simple and non-informative method. This initial estimate will be used to calibrate the first mixture model and then the iterative process is launched. This first estimate of the traffic matrix should not rely on strong statistical assumptions on the OD pairs since arbitrary assumptions will possibly do more harm than good. For these reasons, we have considered two very simple methods : the gravity model [18] and a method based on the second moments of the traffic flows [19]. These methods have been explained respectively in the section 4 and 3.2.

7.4.2 Smoothing the exchange of information between boxes

In the first iterations of the global loop the estimates of the traffic matrix produced by the MCMC samplers as well as the distributions produced by the EM algorithm are not very reliable. These quantities are nevertheless exchanged between the two boxes.

If the exchanges of information were too strong during the first iterations, then the algorithm could converge to a fixed point that would not reflect the actual distribution of the OD counts, as this information is possibly unreliable. It is therefore convenient to "smooth" the information exchanged between boxes during the first iterations properly. This is done as follows :

(i) The prior distributions $p(x_i^t)$ (OD pair i, time t) that are fed back into the first box are not reliable in the first iterations. To smooth these priors we replace $p(x_i^t)$ by $[p(x_i^t)]^\alpha$ where $\alpha < 1.0$. In practice, one only has to use Equation (15) instead of Equation (6) :

$$p(x_2^i \mid \mathbf{y}, \mathbf{x}_2^{-i}) \leftarrow [p(x_2^i)]^\alpha \prod_{i=1}^{r} [p(x_1^i)]^\alpha \tag{23}$$

where the scaling factor is $\alpha < 1.0$.

(ii) Similarly, the estimated OD pairs $\hat{x}_i^t$ produced by the Metropolis within Gibbs algorithm are not reliable in the first iterations. These estimates are nevertheless fed as inputs to the second box. It is therefore convenient to smooth this exchange as well. As the quantity $\hat{x}_i^t$ is used in the EM algorithm only through its likelihoods $G_{\mu,\sigma}(\hat{x}_i^t)$ under the various Gaussian component distributions, smoothing is obtained by raising these likelihoods to the power α in the Expectation step of the EM algorithm (Equation (8)) :

$$G_{\mu,\sigma}(x_i^t) \leftarrow [G_{\mu,\sigma}(\hat{x}_i^t)]^\alpha \tag{24}$$

where $\alpha < 1.0$.

8 Simulation results

The algorithm that we propose has already been validated on simulated data [14][12]. In the present paper, the algorithm is validated on real traffic data. The traffic measurements were performed on a single router network for which direct measurements of the OD counts were made available through special software. They were used by Cao et al as a real dataset to test the validity of their algorithm [13][2] and they were made available to us by the authors.

On this simple network, there are 30 Origin Destination pairs and 10 independent "links". To check our simulations only the Origin Destination pairs that represent more than one percent of the total traffic were considered. The other OD pairs play the role of some adjustment variables.

We found out by simulation that the exchange of information between the two boxes of Figure 7.2 improves the estimate of the traffic matrix along the successive iterations, as it was expected. This can be

seen from Figure 8 which represents the correlation between the true OD pair and its estimated counterpart along the successive iterations of the global loop.

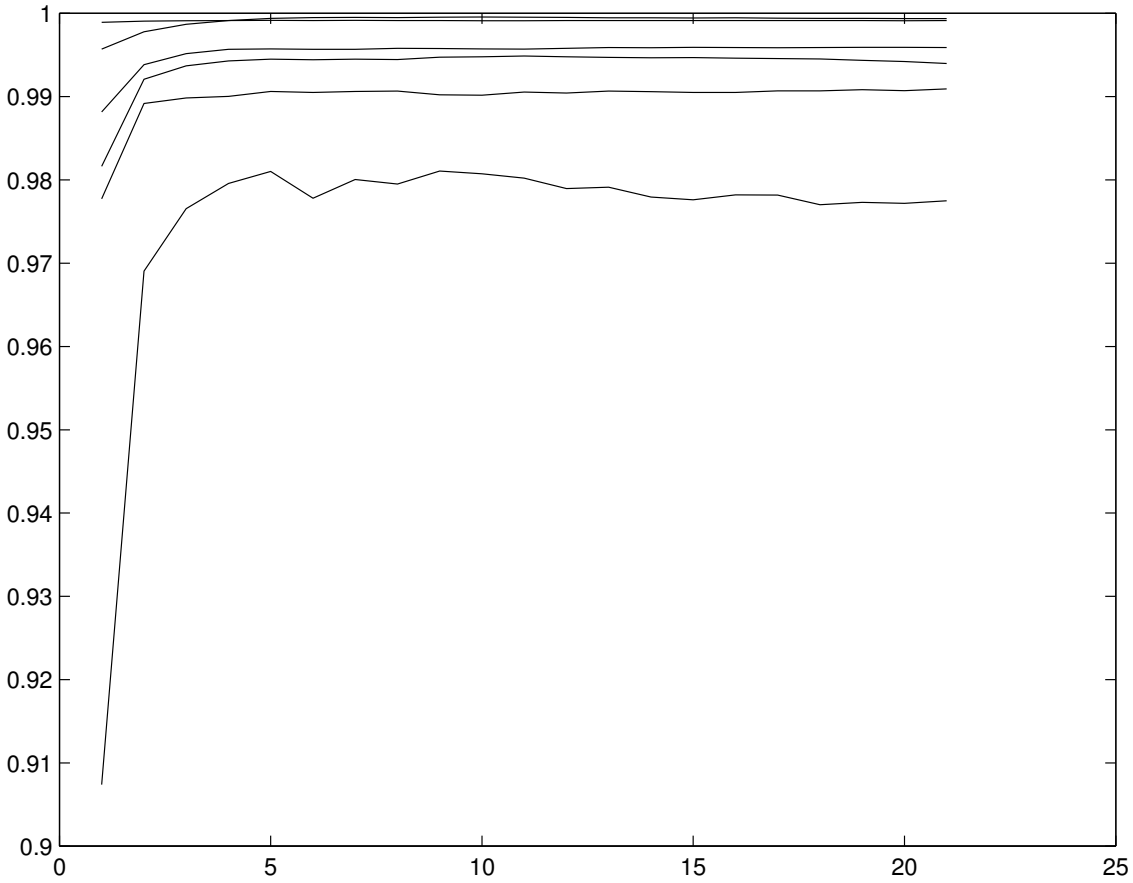


Figure 4: x-axis: iteration number. y-axis: correlation between true and estimated OD pairs.

Then we compared the performance of our algorithm with that of some other algorithms. Various algorithms were tested : EM algorithm with a Poisson OD flows model (Vardi, [10]), a method based on the 1st and 2nd moments (Vanzwet, [19]) and the gravity model. Moreover, for the last two algorithms, we have also tested the performance when the estimator is projected over the space $\{\mathbf{x}_t; \mathbf{y}_t = \mathbf{A}\mathbf{x}_t\}$.

We compared the estimated OD volumes $\hat{x}_t^i$ with the OD volumes x_t^i measured directly on the router by a special software. The Root Mean Square Error $RMSE(i) = [(1/T)\sum_{t=1}^{T}(\hat{x}_t^i - x_t^i)^2]^{1/2}$ was computed for each OD pair i and each algorithm. The results are given in Table 2. In this Table the OD flows are sorted by order of decreasing average volume, so that the first lines correspond to the most important flows. The RMSE values are expressed in terms of a proportion of the average volume for each OD couple.

Thus, one can read in line 1 that the RMSE is 15.2% for the OD pair number 1. For the same OD pair, the RMSE is respectively equal to 15.4%, 21%, 57.3%, 33.5%, 246.8% for respectively the method based on the 1st and 2nd order moments (with a projection), the gravity method (with a projection), the gravity method (without a projection), the method based on the 1st and 2nd order moments (without a projection) and the EM algorithm with Poisson assumption. The results for the other OD pairs can be read in Table 2. An average performance on the whole set of OD pairs is given in the last line of Table 2. The average performance was calculated by giving to the performance on OD pair i a weight proportional to the average volume of this pair. As it can be read in the last line of Table 2, the average RMSE of our method is 24.7%,

OD pair	Throughput			Root Mean Square Error / Average Throughput						
	average, kbits/sec	peak, kbits/sec	(%)	Vardi	Vardi +init.	2nd moment	Gravity	Gravity +proj.	2nd moment +proj.	Vaton et al
1	66 895	1 017 829	31,3	2,663	2,468	0,335	0,573	0,210	0,154	0,152
2	52 724	706 236	24,7	1,482	1,102	0,704	0,625	0,252	0,235	0,174
3	34 440	282 345	16,1	1,616	1,269	0,609	0,700	0,525	0,389	0,398
4	25 333	295 207	11,9	2,006	1,742	1,445	0,477	0,314	0,436	0,222
5	24 706	667 527	11,6	2,777	2,590	0,301	0,883	0,635	0,384	0,205
6	5 802	161 470	2,7	2,045	2,324	1,870	1,117	0,798	0,885	0,530
7	2 022	50 396	0,9	0,667	1,418	2,835	1,471	2,028	2,063	1,484
8	418	178 865	0,2	0,934	15,870	3,130	2,174	1,934	2,654	2,324
9	356	9 264	0,2	0,106	1,859	3,142	1,969	1,497	1,975	1,051
10	236	43 084	0,1	0,240	7,167	4,072	1,917	1,199	1,255	1,125
11	134	4 399	0,1	0,044	2,105	3,205	1,792	4,446	3,241	1,408
12	97	72 794	0,0	4,836	19,806	1,845	1,569	2,743	2,856	3,615
13	79	403	0,0	0,009	0,773	5,353	0,776	2,922	3,352	5,624
14	60	129	0,0	9,396	0,195	2,193	0,961	1,308	1,641	1,066
15	59	7 905	0,0	0,076	9,093	1,997	1,783	3,080	1,823	1,805
16	39	341	0,0	0,014	2,139	4,622	2,161	3,665	3,640	4,384
17	1	5	0,0	7,412	0,469	4,292	1,390	1,344	1,439	1,000
18	1	1	0,0	0,284	0,249	0,987	1,029	2,311	2,489	2,667
19	1	3	0,0	0,000	0,511	4,176	2,110	4,379	3,976	1,189
20	0	3	0,0	0,000	0,812	2,526	1,198	1,502	1,751	1,486
Overall error ~>				2,091	1,890	0,684	0,663	0,379	0,325	0,247

Table 2: performance comparison of various OD matrix estimation algorithms.

whereas it is equal to 32.5% for its main "competitor" (method based on the 1st and 2nd order moments with a projection). Thus, for this dataset, our method improves the RMSE by 8% with respect to the main competitor. The other methods have an average RMSE of respectively 37.9%, 66.3%, 68.4%, 189%, for respectively the gravity method with a projection, the gravity method without a projection, the method based on the first and second moments without a projection, the EM algorithm with Poisson model. As one can see, it is remarkable that the projection improves considerably the estimators as noticed in [18].

On Figure 5 the variations of one OD pair along the time are displayed. The variations are displayed over 5 consecutive days with a time granularity of 5 minutes, that is to say that each dot represents the total volume of traffic for this OD pair during a period of 5 minutes. This makes an overall time series of 1440 successive values for that OD pair. The "true" values of the OD volume (obtained by measuring directly the OD volumes with a special software) are represented as squares, whereas the values that we obtain with our algorithm are represented as stars. As one can see from this Figure the deviation of the estimated values from the true values is quite small, which is conformant with the low RMSE (15.2% for this OD pair).

A reader experienced in time series analysis could see from this Figure that a hidden Markov model or a mixture model seems quite relevant for that find of traffic. It is therefore not surprising that our estimate is betterr than others for this dataset since our algorithm takes a mixture model explicitely into account for each Origin Destination couple. The same results have been displayed on Figure 6 for another OD pair, namely the OD pair number 5. Once again, for this OD pair, the estimation is pretty accurate. Similar plots could be provided for the other OD pairs iof that network.

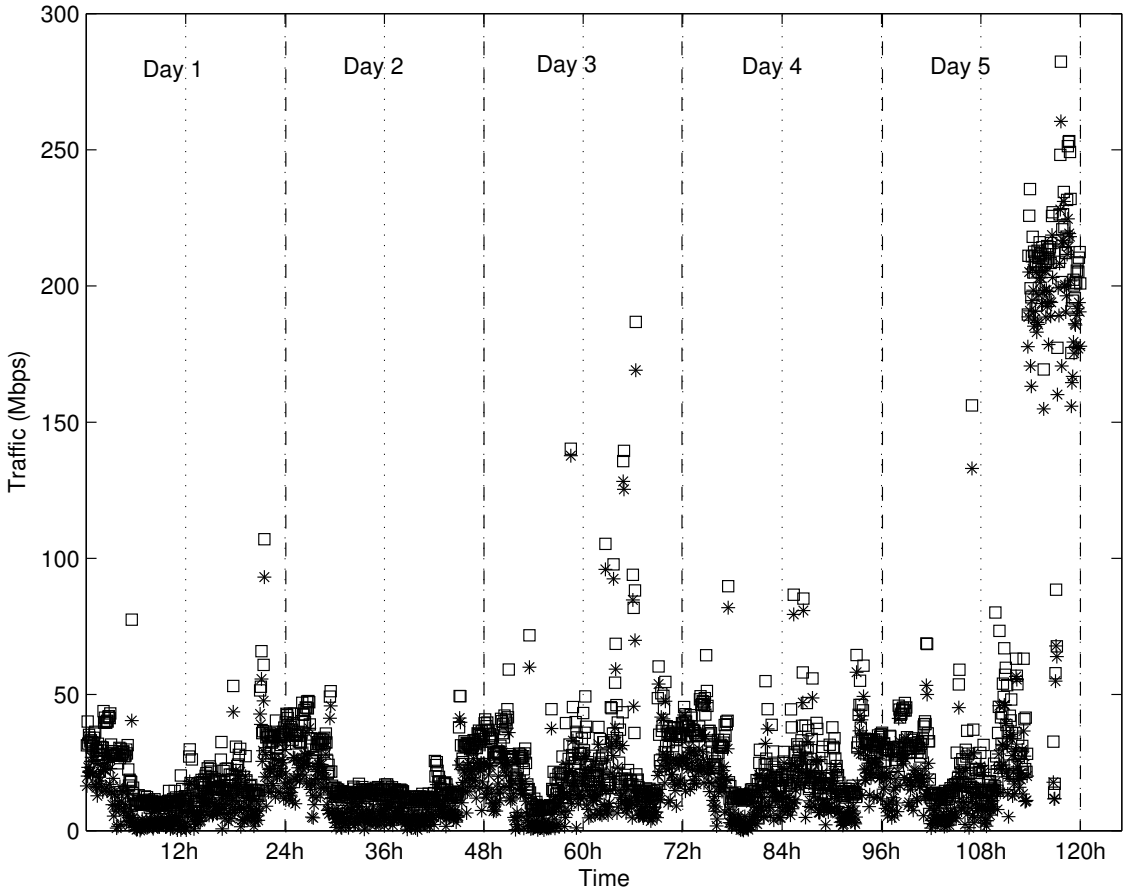


Figure 5: OD pair number 1. 5 days of traffic with a granularity of 5 minutes. Comparison of the measured values of the OD volume with the values produced by our estimation algorithm.

9 Conclusion

Markov Chain Monte Carlo methods are flexible and reliable techniques that can be used to estimate the traffic matrix on a network. One major drawback of these methods is that they require reliable knowledge of a priori distribution for the OD pairs and that, most of the time, reliable prior distributions are not available.

In this paper, we have proposed a method for training these prior distributions from the only available data, that is to say the link counts themselves. Furthermore, we improve our priors in a loop by taking into account both time and OD pairs dependencies.

All the steps of our method are explained in detail and every arbitrary choice has been underlined. We have exposed our scientific path through the methods of estimation of the traffic matrix. All the initial values and adjustments of the algorithms used have been shown. As a result, one should easily recover our results with his own program if one follows our instructions.

We have used our method on real traffic data on a simple network but with a great burstiness. The results are exposed in this paper and compared to other methods of traffic matrix estimation. This confirms the positive results that we had obtained on simulated data in previous papers [15] [13].

However, the field of investigation remains large. Indeed, we should go further in the exploration of the bounds of precision one can reach in the problem of traffic matrix estimation. This could lead us to identify the informations which could help us a lot to know more about the interesting traffics. As a consequence,

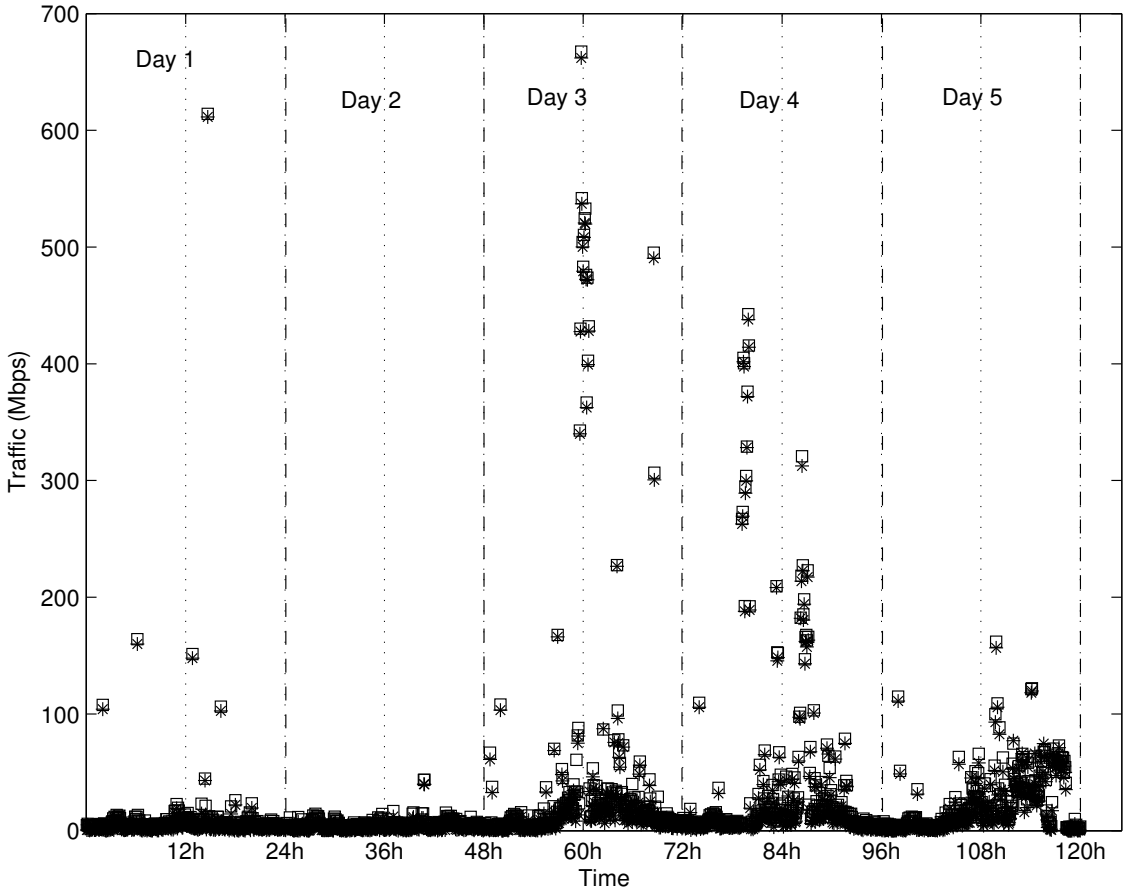


Figure 6: OD pair number 5. 5 days of traffic with a granularity of 5 minutes. Comparison of the measured values of the OD volume with the values produced by our estimation algorithm.

operators could fully take benefit of new technologies like BGP accounting which allow us to have more measures on smaller aggregates than links between routers.

Another interesting subject is the routing matrix. Our method, like lots of others, suppose that the routing matrix is constant during all the experiment. But in real backbone networks, stability is not always possible. That is why we should try to adapt our thoughts to a more general framework, that is to say with a changing routing matrix.

In addition, we should explore new methods for moving faster and better than with Metropolis within Gibbs into the space of traffic matrices which satisfy the routing constraints given the link counts. Indeed, a really important improvement of our algorithm would be a decrease in its complexity since it is for now unfeasible for big backbone networks with lots of nodes.

References

[1] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian, "Traffic Matrix Estimation: Existing Techniques Compared and New Directions," in *SIGCOMM*, Pittsburgh, Aug. 2002.

[2] J. Cao, D. Davis, S. Vander Wiel, and B. Yu, "Time-Varying Network Tomography: Router Link Data," *Journal of the American Statistical Association*, vol. 95, no. 452, Dec. 2000.

[3] V. Paxson and S. Floyd, "Wide-Area Traffic : the Failure of Poisson Modelling," *IEEE/ACM Trans. on Networking*, 1995.

[4] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Spinger, 2001.

[5] G.H. Golub and C.F. Van Loan, *Matrix Computations, Third Edition*, Johns Hopkins Series in Mathematical Sciences, 1996.

[6] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, "How to Identify and Estimate the Largest Traffic Matrix Elements in a Dynamic Environment," in *ACM Sigmetrics*, June 2004.

[7] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An Information Theoretic Approach to Traffic Matrix Estimation," in *ACM SIGCOMM*, 2003.

[8] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley Series in Probability and Statistics, 1997.

[9] L.R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. IEEE, 1989.

[10] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities From Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, March 1996.

[11] C. Tebaldi and M. West, "Bayesian Inference on Network Traffic Using Link Count Data," *Journal of the American Statistical Association*, vol. 93, no. 442, June 1998.

[12] S. Vaton and A. Gravey, "Network Tomography : an Iterative Bayesian Analysis," in *ITC 18th*, Berlin, Aug.-Sep. 2003.

[13] J. Cao, S. Vander Wiel, B. Yu, and Z. Zhu, "A Scalable Method for Estimating Network Traffic Matrices from Link Counts," Tech. Rep., Bell Labs, 2000.

[14] S. Vaton and A. Gravey, "Iterative Bayesian Analysis of Network Traffic Matrices in the Case of Bursty Flows," in *Internet Measurement Workshop*, Marseille, Nov. 2002.

[15] D.M. Titterington, A.F.M. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley and Sons, 1985.

[16] N. Duffiels Y. Zhang, M. Roughan and A. Greenberg, "Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads," in *ACM SIGMETRICS*, San Diego, Jun. 2003.

[17] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data," *JRSS B*, vol. 39, pp. 1–38, 1977.

[18] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in Measuring Backbone Traffic Variability : Models, Metrics, Measurements and Meaning," in *Internet Measurement Workshop*, Marseille, Nov. 2002.

[19] E. Van Zwet, "Method of Moments Estimation for Origin-Destination Traffic on a Network," Dept of Statistics, University of California in Berkeley.

Quick Traffic Matrix Estimation Based on Link Count Covariances

Ilmari Juva[1], Sandrine Vaton[2], Jorma Virtamo[1]
1: Helsinki University of Technology, Finland, 2: ENST Bretagne, France
ilmari.juva@tkk.fi sandrine.vaton@enst-bretagne.fr jorma.virtamo@tkk.fi

***Abstract*— In this paper we consider the problem of traffic matrix estimation. As the problem is underconstrained, some additional information has to be brought in to obtain a solution. If we have a sequence of link count measurements available, a natural candidate is to use the link count sample covariance matrix under the assumption of a functional relationship between the mean and the variance of the traffic. We propose two computationally light-weight methods for traffic matrix estimation based on the covariance matrix, the projection method and constrained minimization method. The accuracy of these methods is compared with that of other methods using second order moment estimates by simulation under synthetic traffic scenarios.**

Keywords: Traffic Matrix Estimation

I. Introduction

The traffic matrix, which gives volume of traffic between each origin/destination (OD) pair in the network, is a required input in many network management tasks. Such tasks include for instance traffic engineering and network capacity dimensioning. In many cases the knowledge on the underlying traffic volumes is assumed to be known. It is recognized that accurate demand matrices are crucial for traffic engineering. However, in reality, they are seldom available in current IP networks.

This raises the need to estimate the traffic matrix based on information readily available. Traffic matrix inference usually utilizes SNMP measurement data. While it has some limitations, the attractive feature of SNMP is that it is usually available everywhere in an IP network and is the only widespread tool to obtain link count data. We denote a set of link count measurements by $\boldsymbol{y}$. The routing matrix $\boldsymbol{A}$ is also readily available.

In traffic matrix estimation, the basic relationship between link counts $\boldsymbol{y}$ and origin-destination counts $\boldsymbol{x}$ can be written as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{A}$ is the routing matrix and $\boldsymbol{x}$ is the OD counts in vector form, i.e. each component represents a traffic of one OD pair. The above equation holds exactly and also the equation where we take the expectations from $\boldsymbol{y}$ and $\boldsymbol{x}$ holds. The expected valued of $\boldsymbol{x}$ is the traffic matrix $\boldsymbol{\lambda}$, and we get the first moment equation of traffic matrix estimation

$$\overline{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\lambda}, \tag{2}$$

where $\overline{\boldsymbol{y}}$ denoted the sample mean of the link counts.

Since in any realistic network there are many more OD pairs than links, the problem of solving $\boldsymbol{\lambda}$ from $\boldsymbol{A}$ and $\boldsymbol{y}$ is strongly underdetermined. This means that accurate explicit solutions cannot be found, as there is an infinite number of solutions for $\boldsymbol{\lambda}$ that satisfy equation (2). To overcome this ill-posedness, some type of additional information has to be brought in to solve the problem. Reviews of the proposed methods can be found e.g. in [1], [2] and [3].

Typically, methods that need a prior distribution use the gravity model to obtain one. However, the gravity assumption that the traffic volume of an OD pair is proportional to the total traffic sent by the origin node and the total traffic terminating at the destination node, does not always hold. In [4] the authors study real traffic matrix of a North American backbone network and conclude that there are significant errors concerning the estimation of the largest OD pairs, which are the most important ones for traffic engineering purposes. Therefore, in this paper we propose another way of obtaining prior distribution based on the link count covariances and a functional mean-variance relationship. Based on this, we develop two computationally lightweight methods, similar in principle to the tomogravity method of [5], in the sense that they incorporate a starting point and link count measurements to obtain an estimate.

The rest of the paper is organized as follows. In section II we discuss methods relying on the link covariance as the additional information. In section III we show how to solve OD pair covariance matrix from the link count covariance matrix. Section IV and V present our quick estimation methods, the projection method and the constrained minimization method. In section VI we compare the performance of these methods to maximum likelihood estimation. And finally section VII concludes the paper.

II. Methods based on link count covariances

Maximum likelihood estimation (MLE) uses the second moment statistic, the link count covariance, as the additional information that is needed to yield an estimate. It is also necessary to assume local stationarity for the measurements considered, and a distribution which the stochastic fluctuation of the traffic follows.

In [6], Vardi first proposed this kind of approach. The Poisson distribution is assumed, meaning that variance is equal to the mean, and the system becomes

$$\begin{pmatrix} \overline{\boldsymbol{y}} \\ \epsilon \boldsymbol{S}^{(y)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} \\ \epsilon \boldsymbol{B} \end{pmatrix} \boldsymbol{\lambda}, \tag{3}$$

where, as is explained in more detail in section III, $\boldsymbol{S}^{(y)}$ is the sample link covariance matrix and $\boldsymbol{B}$ is the matrix of element-wise products of rows of $\boldsymbol{A}$. Coefficient $\epsilon \in (0, 1]$ defines how much weight is given to the second moment estimate in the final solution, and $\boldsymbol{\lambda}$ is the estimator for the mean of $\boldsymbol{x}$. This is a linear inverse positive, or LININPOS, problem and can be solved by numerical likelihood methods, such as the EM-algorithm. The solution obtained this way minimizes the Kullback-Leibler distance between the observed moments and theoretical values. If we instead minimize (3) in least square sense, the solution is easily obtained in closed form.

Vardi's method, however, does not give very accurate estimates, as was discovered in [2] and [4]. This is due to the fact that the Poisson assumption is not accurate in current IP networks. Cao et al. [7] generalize the maximum likelihood approach by assuming a Gaussian traffic distribution and assuming that the variance is related to the mean through a power-law. While this MLE approach is efficient and theoretically justifiable, the size of the problem in traffic matrix estimation requires the use of iterative numerical methods, such as the Expectation Maximization algorithm, which is computationally quite heavy.

The MLE relies on the fact that the system of first and second order link count statistics together make the system identifiable with regard to the first order OD-pair statistics, i.e. we are able to find solution for the likelihood equations if there exists a functional relationship between the mean and the variance of OD-pair traffic. The commonly used relation is the power-law relation

$$\boldsymbol{\Sigma} = \phi \cdot \mathrm{diag}\{\boldsymbol{\lambda}^c\}, \tag{4}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix, because we assume independence between OD pairs.

But, in fact, the second order statistic for OD-pairs is identifiable based solely on the second order statistic of the link counts, as long as we assume independence among OD-pairs and a sensible routing scheme. This result is proven by Soule et al. [8]. Since we can analytically solve the variance of the OD-pairs by least squares method, and the power-law relation between variance and mean is assumed, we can then solve the traffic matrix from our variance estimate.

The benefit is that this does not call for numerical methods, and is thus extremely quick to calculate. The problem with this approach is that it does not take into account the first moment equation 2, which is a stronger condition as opposed to the mean-variance relation which is only an assumption. Our methods incorporate this information into the solution obtained through estimation of the variance yet maintaining the computational simplicity of the model.

III. Solving OD-pair covariance matrix from link counts

Let us denote the number of links by J and the number of OD-pairs by N. Then the vector form of traffic matrix $\boldsymbol{x}$ has the dimension $(N \times 1)$, link loads $\boldsymbol{y}$ has the dimension $(J \times 1)$.

First, let us define $\boldsymbol{S}^{(y)}$ as a $\frac{1}{2}J(J+1)$-vector containing diagonal and upper triangle elements of the link covariance matrix $\boldsymbol{\Sigma}^{(y)}$. Define $\boldsymbol{S}^{(x)}$ as an N-vector containing the diagonal elements of the OD-pair covariance matrix $\boldsymbol{\Sigma}^{(x)}$. $\boldsymbol{A}$ is the $(J \times N)$ routing matrix, whose element $A_{i,j}$ is 1 if OD pair x_j uses link y_i, and 0 otherwise. Then define a $(\frac{1}{2}J(J+1) \times N)$ matrix $\boldsymbol{B}$ that relates vector $\boldsymbol{S}^{(y)}$ to vector $\boldsymbol{S}^{(x)}$. A row of $\boldsymbol{B}$ is indexed by a compound index (ij) where $i = 1, \ldots, J;\ j = i, \ldots, J$, meaning that the index runs through $\frac{1}{2}J(J+1)$ values,

$$B_{(ij),k} = A_{i,k}A_{j,k} \qquad \begin{array}{l} i = 1, \ldots, J;\ j = i, \ldots, J \\ k = 1, \ldots, N. \end{array}$$

In vector form this reads,

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{A}_1 \star \boldsymbol{A}_1 \\ \boldsymbol{A}_1 \star \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_1 \star \boldsymbol{A}_c \\ \boldsymbol{A}_2 \star \boldsymbol{A}_2 \\ \boldsymbol{A}_2 \star \boldsymbol{A}_3 \\ \vdots \\ \boldsymbol{A}_c \star \boldsymbol{A}_c \end{pmatrix},$$

where $\boldsymbol{A}_i$ denotes the ith row of $\boldsymbol{A}$, and the componentwise product is denoted with the star ($\star$). Now the rows of $\boldsymbol{B}$ indicate the elements of $\boldsymbol{x}$ contributing to covariance between links i and j.

The measured link covariance matrix can be written as

$$\boldsymbol{\Sigma}^{(y)} = \sum_k \sigma_k^2 \boldsymbol{a}_k \boldsymbol{a}_k^{\mathrm{T}}, \tag{5}$$

where $\boldsymbol{a}_i$ is the ith column of $\boldsymbol{A}$. In component form we have

$$\boldsymbol{\Sigma}^{(y)}_{i,j} = \sum_k \sigma_k^2 A_{i,k} A_{j,k}. \tag{6}$$

Using vector notation, the equation becomes

$$\boldsymbol{S}^{(y)} = \boldsymbol{B}\boldsymbol{S}^{(x)}. \tag{7}$$

This is in fact quite similar to (3) in the case where ϵ would be set very large, leading to the part $\boldsymbol{S}^{(y)} = \boldsymbol{B}\boldsymbol{\lambda}$ to dominate the equation. We just have a more general power-law relation instead of the Poisson assumption, so we cannot now just replace $\boldsymbol{S}^{(x)}$ with $\boldsymbol{\lambda}$.

Typically $\frac{1}{2}J(J+1) > N$ and equation (7) is overdetermined. The least square estimate (LSE) solution (see e.g. [9]), to the equation is

$$\boldsymbol{S}^{(x)} = (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}. \tag{8}$$

IV. Projection method

Now that we have an estimate for the variances of each OD-pair, it is trivial to find an estimate of the mean by using the mean-variance relation (4).

$$\boldsymbol{\lambda}_0 = (\phi^{-1}\boldsymbol{S}^{(x)})^{\frac{1}{c}}. \tag{9}$$

The problem with this estimate is, that it does not require the solution to satisfy the link count equation (2), which is a stronger condition than the second moment relation. The preliminary estimate $\boldsymbol{\lambda}_0$ can be improved by projecting the result to the surface that satisfies the first moment condition. This yields our estimate

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\lambda}_0). \tag{10}$$

Compared to the maximum likelihood approach, we do the moment estimation sequentially: First obtaining an estimate for the covariance matrix and then solving for the mean. This does not yield quite as accurate estimates as MLE, but it is many times faster.

The projection might yield negative values for smaller OD pairs, as no positivity constraint is imposed in order to keep the method as light-weight as possible. We simply substitute the negative estimates by zero, concluding that these OD pairs are negligibly small.

The projection method works for any fixed parameters ϕ and c. In fact, we can try to estimate these parameters by requiring that $\boldsymbol{\lambda}_0$ comes as close as possible to satisfying (2), i.e. that they minimize

$$\begin{aligned} f(\phi, c) &= (\overline{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{\lambda}_0)^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{\lambda}_0) \\ &= (\overline{\boldsymbol{y}} - \boldsymbol{A}(\phi^{-1}\boldsymbol{S})^{\frac{1}{c}})^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}(\phi^{-1}\boldsymbol{S})^{\frac{1}{c}}). \end{aligned} \tag{11}$$

The values of ϕ and c that realize the minimum, can now be used in equation (9) to yield the estimate $\boldsymbol{\lambda}$.

A. Estimating the parameters ϕ and c

In Cao et al. [7], the EM-algorithm is run after preselecting a convenient value for the exponent parameter c in the power law relation (4), while ϕ remains a parameter that the algorithm optimizes. The authors point out that convergence is guaranteed for the algorithm only for integer values of c, namely 1 or 2. However, Gunnar et al. [4] in their study of the Global Crossing data find out that the correct values for c in those particular networks are 1.5 and 1.6 for the European and North American core-networks respectively. Thus being limited to integer values in the solution makes sense for only computational reasons. The projection method, on the other hand, works for any preselected c. And, in fact, we can relax c to be a free parameter, though this means that we will no longer be able to obtain a closed form solution.

Minimization of (11) with respect to ϕ is a simple quadratic problem. So we can easily find the minimizing value $\widehat{\phi}(c)$. Now we can either use a preselected value for c to yield the optimal ϕ value, or insert $\widehat{\phi}(c)$ back to (11), which yields

$$f(\widehat{\phi}(c), c) = (\overline{\boldsymbol{y}} - \boldsymbol{A}(\widehat{\phi}(c)^{-1}\boldsymbol{S})^{\frac{1}{c}})^{\mathrm{T}}(\overline{\boldsymbol{y}} - \boldsymbol{A}(\widehat{\phi}(c)^{-1}\boldsymbol{S})^{\frac{1}{c}}). \tag{12}$$

Now we have a simple one parameter numerical optimization to find the optimal value of c. Expression (12) as a function of c is depicted in Figure 1. The figure is based on a set of synthetic data that was generated by using the parameter value $c = 1.5$. Figure 2 shows a histogram of estimated values for the parameter c.

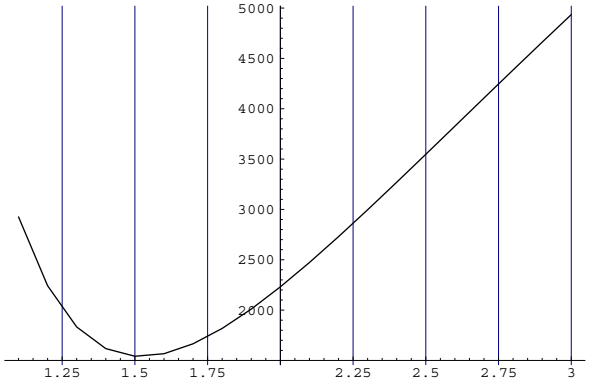


Fig. 1. Values of the objective function (12) as a function of parameter c.

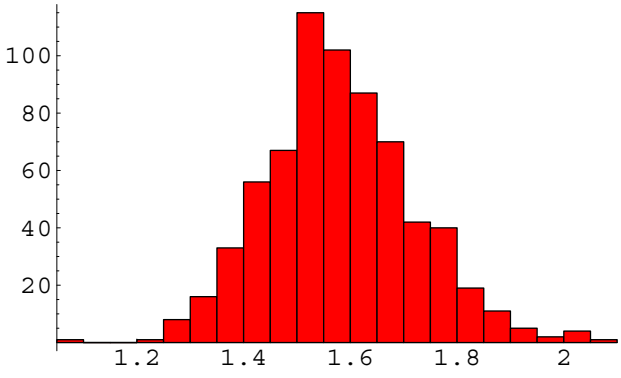


Fig. 2. Estimated c-values for synthetic data sets of size 500, generated with setting $c = 1.5$.

V. Constrained minimization

Another approach is to require the condition $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\lambda}$ to be satisfied from the outset, and try to satisfy the mean-variance relation in the least square sense. In general, this has to be solved numerically. However, in the special case of $c = 1$ an explicit solution can be derived.

This approach is equivalent to Vardi's method, if we set ϵ very small so that the first moment is the dominant factor in the estimation, with the exception that we treat ϕ as a parameter to be optimized, whereas in (3) it is fixed as 1 by the Poisson assumption.

We get a constrained minimization problem

$$\begin{aligned} \min_{\boldsymbol{\lambda},\phi} \quad & \|\boldsymbol{S}^{(y)} - \boldsymbol{B}\phi\boldsymbol{\lambda}^c\| \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{\lambda}. \end{aligned} \tag{13}$$

Introducing a vector of Lagrange multipliers $\boldsymbol{\alpha}$, the objective function to be minimized can be written as

$$\begin{aligned} & f(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \phi) \\ &= (\boldsymbol{S}^{(y)} - \phi\boldsymbol{B}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{S}^{(y)} - \phi\boldsymbol{B}\boldsymbol{\lambda}) + 2\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\lambda}) \\ &= \phi^2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda} - 2\phi{\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda} - 2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{\lambda} \\ &\quad + {\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{S}^{(y)} + 2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{y}. \end{aligned} \tag{14}$$

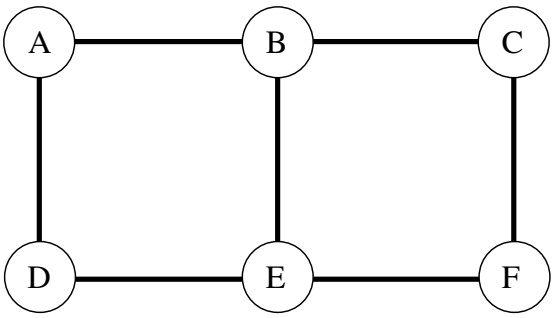


Fig. 3. Six node Test topology

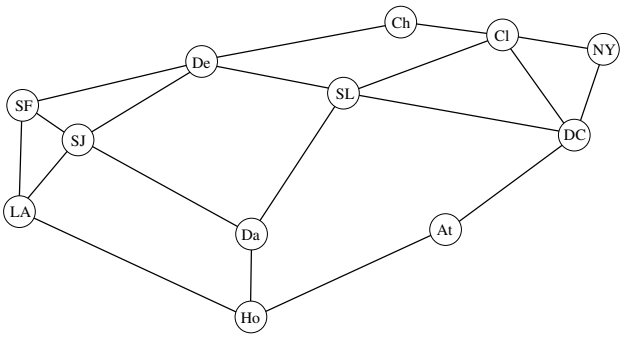


Fig. 4. Twelve node backbone test topology

The above expression is quadratic in $\boldsymbol{\lambda}$, and the minimum with respect to $\boldsymbol{\lambda}$ can easily be found,

$$\boldsymbol{\lambda} = \phi^{-2}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\alpha} + \phi\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}). \quad (15)$$

The Lagrange multipliers $\boldsymbol{\alpha}$ are then determined such that the constraints are satisfied:

$$\boldsymbol{y} = \boldsymbol{A}\phi^{-2}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\alpha} + \phi\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}), \quad (16)$$

from which

$$\boldsymbol{\alpha} = (\phi^{-2}\boldsymbol{A}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{A}^{\mathrm{T}})^{-1} \cdot \cdot (\boldsymbol{y} - \phi^{-1}\boldsymbol{A}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{(y)}). \quad (17)$$

Minimizing $f(\boldsymbol{\lambda}, \boldsymbol{\alpha}, \phi)$ with respect to ϕ yields

$$\phi = (\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda})^{-1}{\boldsymbol{S}^{(y)}}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda}. \quad (18)$$

Substitution of (17) into (15) gives $\boldsymbol{\lambda}$ as a function of ϕ

$$\boldsymbol{\lambda} = \boldsymbol{K}\boldsymbol{y} - \phi^{-1}\big(\boldsymbol{K}\boldsymbol{A}(\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{B}^{T}\boldsymbol{S}^{(y)} + \boldsymbol{B}^{T}\boldsymbol{S}^{(y)}\big),$$

where we use the notation

$$\boldsymbol{K} = (\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{A}^{T}(\boldsymbol{A}(\boldsymbol{B}^{T}\boldsymbol{B})^{-1}\boldsymbol{A}^{T})^{-1}.$$

Substituting $\boldsymbol{\lambda}$ further in (18) yields an quadratic equation for ϕ, which is easily solvable. This solution can be then substituted back to (17) and (15) to obtain the explicit expression for $\boldsymbol{\lambda}$.

VI. Comparison with the MLE method

The accuracy of the quick methods are evaluated by comparing them against Maximum likelihood estimation. In the following subsection we present the Maximum likelihood estimation used. In the subsequent sections the results of accuracy on synthetic data test cases is presented.

A. Maximum Likelihood Estimation

We follow the approach of Cao et al. [7] in using the Expectation Maximization (EM) algorithm. For a review see also [2].

The log-likelihood for estimating $\boldsymbol{\lambda}$ is given as

$$l(\boldsymbol{\theta}|\boldsymbol{Y}) = -\frac{\tau}{2}\log|\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}| - \quad (19)$$
$$-\frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda}),$$

where τ is the number of measurements and $\boldsymbol{y}_t$ is the link count vector for measurement t. In Cao et al. c is assumed to be constant and the parameters of the model are thus

$$\boldsymbol{\theta} = (\phi, \boldsymbol{\lambda}).$$

We can write $\boldsymbol{\Sigma}$ as a function of $\boldsymbol{\theta}$ according to (4). Now the problem can be solved numerically with the EM-algorithm. The complete data log-likelihood is of the form

$$l(\boldsymbol{\theta}|\boldsymbol{X}) = -\frac{\tau}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{x}_t - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \boldsymbol{\lambda}).$$

The EM-equation is

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \mathrm{E}[l(\boldsymbol{\theta}|\boldsymbol{X})|\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}]$$
$$= \mathrm{E}[-\frac{\tau}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{x}_t - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \boldsymbol{\lambda})|\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}]$$

Since

$$\begin{aligned}
&\mathrm{E}[(\boldsymbol{x} - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\lambda})] \\
&= \mathrm{E}[\mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\lambda})(\boldsymbol{x} - \boldsymbol{\lambda})^{\mathrm{T}}\}] \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\mathrm{E}[(\boldsymbol{x} - \boldsymbol{\lambda})(\boldsymbol{x} - \boldsymbol{\lambda})^{\mathrm{T}}]\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\mathrm{E}[((\boldsymbol{x} - \boldsymbol{m}) + (\boldsymbol{m} - \boldsymbol{\lambda}))((\boldsymbol{x} - \boldsymbol{m}) + (\boldsymbol{m} - \boldsymbol{\lambda}))^{\mathrm{T}}]\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{R} + (\boldsymbol{m} - \boldsymbol{\lambda})(\boldsymbol{m} - \boldsymbol{\lambda})^{\mathrm{T}})\} \\
&= \mathrm{Tr}\{\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\} + (\boldsymbol{m} - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{m} - \boldsymbol{\lambda})
\end{aligned}$$

we can write

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) &= -\frac{\tau}{2}(\log|\boldsymbol{\Sigma}| + \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{R}^{(k)})) \\
&\quad -\frac{1}{2}\sum_{t=1}^{\tau}(\boldsymbol{m}_t^{(k)} - \boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}_t^{(k)} - \boldsymbol{\lambda}),
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{m}_t^{(k)} &= \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{\theta}^{(k)}] \\
&= \boldsymbol{\lambda}^{(k)} + \boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{\lambda}) \\
\boldsymbol{R}^{(k)} &= \mathrm{Var}[\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{\theta}^{(k)}] \\
&= \boldsymbol{\Sigma}^{(k)} - \boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{A}\boldsymbol{\Sigma}^{(k)}.
\end{aligned}$$

According to [7], convergence to the maximum likelihood estimate is guaranteed in the special cases of $c = 1$ and $c = 2$.

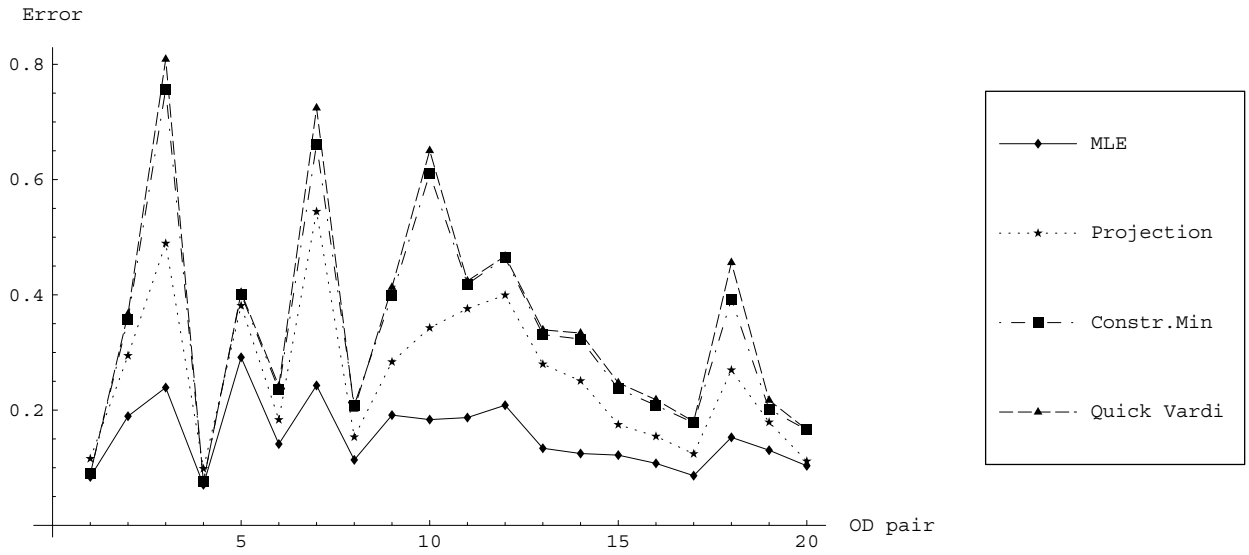


Fig. 5. Errors for OD pairs in 6-node topology in ascending order of traffic amount for case $c = 1$.

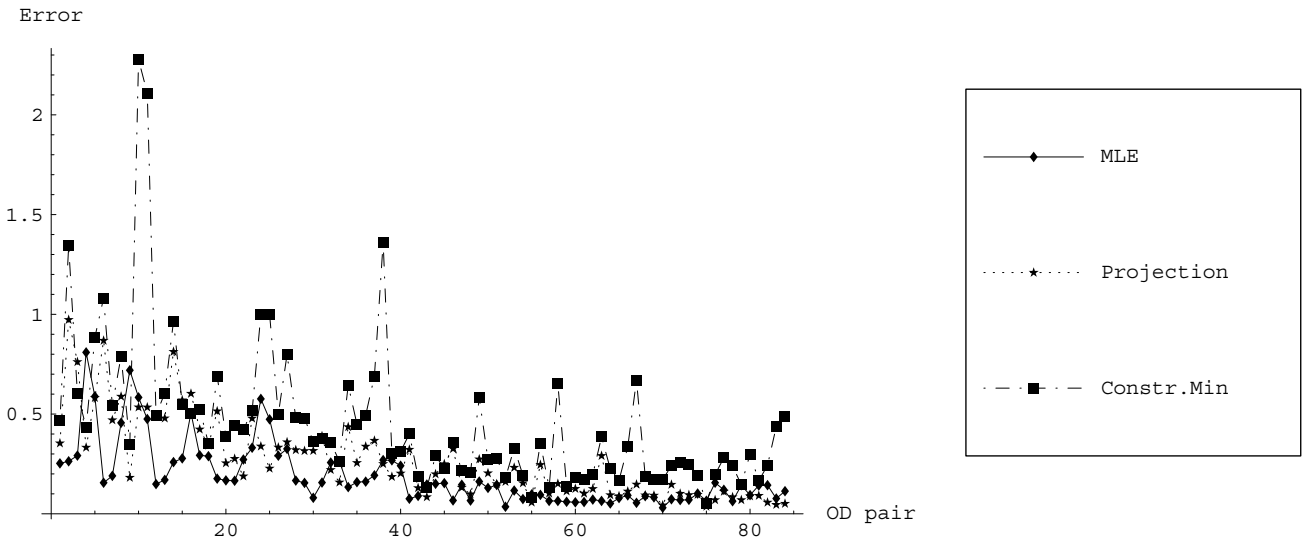


Fig. 6. Errors for largest OD pairs in 12-node topology in ascending order of traffic amount

B. Results

For the evaluation of the methods we use two topologies. A small six-node topology, shown in Figure 3, has 14 one-way links, two links between each connected pair of node. Assuming traffic from each node to all other nodes, there are 30 OD pairs in the network. In the more realistic size fictitious backbone topology shown in Figure 4, there are 12 nodes, 38 links, and 132 OD pairs. For both topologies, we generate synthetic Gaussian data sets, where the power-law holds. Sample size is set to 500 measurements for each simulation.

1) A Simple six node topology: In our synthetic OD pair traffic the traffic varies so that the largest OD pairs are ten times as large as the smallest ones.

Figure 5 shows the results for the maximum likelihood estimates, projection method, the constrained minimization, and Vardi's method solved with the least square method, which we call here Quick Vardi. The synthetic data used for the evaluations is generated with parameters $c = 1$, $\phi = 1$. This is equivalent to the Poisson assumption made in Vardi's method.

The OD pairs are presented in ascending order based on the traffic volume, so that the smaller OD pairs are on the

left and the largest on the right. We see that, as expected, the MLE performs better on average, but not overwhelmingly so. The mean relative errors are 15%, 26%, 34% and 35% for the MLE, the projection method, constrained minimization and Quick Vardi respectively.

2) A 12 node backbone topology: In this example case we use synthetic data generated with the parameter value $c = 1.5$. The traffic volumes for the OD pairs vary so that the largest are approximately hundred times as large as the smallest ones. This creates great difficulties for the quick methods regarding the estimation of the smaller OD pairs. The estimates of the projection method for the smallest OD pairs are far off the real traffic volumes. Due to the fact that the estimates for some of the smallest OD pairs have errors of several hundred percent, the mean relative error is also affected greatly by these, and is 59% for the projection method and 110% for the constrained optimization, while it is 29% for the MLE. The mean error for the Quick Vardi method is several hundred percent, so it is not considered here.

However, the most important thing is to estimate the largest OD pairs. If we concentrate only on the largest OD pairs that comprise 90% of total traffic in volume, the projection method is more competitive. The errors for these OD pairs are shown in Figure 6. The errors are 27% for the projection method, 46% for the constrained minimization and 19% for the MLE.

VII. Conclusion and Future Work

This paper presented ways to obtain estimate for traffic matrix by explicit calculations utilizing the link count covariance matrix. We illustrated how one can obtain the OD pair traffic variance estimates from empirical link count covariance matrix, and developed computationally light weight methods, the projection method and the constrained minimization method, to obtain an estimate for the traffic matrix based on the link count covariance matrix, in a way that would still be consistent with the link counts.

The constrained minimization method was recognized, in fact, to be a special case of Vardi's method. We give an explicit solution for it in the case $c = 1$ and also obtain an estimate for the second parameter ϕ in the mean-variance relation. For the projection method we have an even simpler and quicker to compute solution. Also in this case we get estimates for the parameters c and ϕ.

We evaluated the accuracy of the methods in a simulation study by comparing them against the maximum likelihood solution by Cao et al., and found that they perform reasonably well, considering that they are much quicker and simpler to calculate than the MLE, which requires the use of an iterative numerical method, namely the EM-algorithm. In the worst case, the errors in the estimate of a traffic matrix element for the largest components given by the quick method were three times as large as those by the MLE method, in many cases they difference was smaller. As for the running time, the difference between the MLE method and quick methods was big. With our non-optimized Mathematica code running the MLE method took of the order of tens of minutes, while the quick methods yielded the result in a few seconds.

In this paper all comparisons were made with synthetic data. Evaluation with real data would be very important to assess the true effectiveness of the methods. For now, we have used in our evaluations a sample size of 500, which may be rather large in comparison to what is available in reality. Accuracy of the estimated covariance matrix with various sample sizes should be studied, as well as the effect the measurement inaccuracies have on the subsequent traffic matrix estimates.

Acknowledgments

The work for this paper was carried out as a part of project Fancy funded by Academy of Finland, of the Specific Joint Research Project on Traffic matrix estimation funded by Euro-NGI, and of the research project on the traffic matrix funded by "region Bretagne" (operation A3C942, program 1042).

References

[1] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions", ACM SIGCOMM, Pittsburg, USA. August 2002.

[2] S. Vaton, J.S. Bedo, A. Gravey, "Advanced methods for the estimation of the Origin Destination traffic matrix", Revue du 25ème anniversaire du GERAD, 2005.

[3] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, Traffic matrices: Balancing measurements, inference and modeling, in SIGMETRICS'05, Banff, Canada, June 2005.

[4] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone – A comparison on real data", IMC'04, Taormina, Italy, 2004.

[5] Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads", ACM Sigmetrics 2003.

[6] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data", Journal of the American Statistical Association, vol. 91, pp. 365–377, 1996.

[7] J. Cao, D. Davis, S. V. Wiel, B. Yu, "Time-varying network tomography," Journal of the American Statistical Association, vol. 95, pp. 1063–1075, 2000.

[8] A. Soule, A. Nucci, R. Cruz, E. Leonardi, N. Taft, "How to Identify and Estimate the Largest Traffic Matrix Elements in a Dynamic Environment" SIGMETRICS / Performance'04, New York, USA, pp. 73–84, 2004.

[9] David G. Luenberger, "Optimization by Vector Space Methods", New York NY, John Wiley & Sons, 1969.

Search for Optimality in Traffic Matrix Estimation: a Rational Approach by Cramér-Rao Lower Bounds

Paola Bermolen[1], Sandrine Vaton[2] and Ilmari Juva[3]
paola@fing.edu.uy sandrine.vaton@enst-bretagne.fr ilmari.juva@tkk.fi
1: Universidad de la Republica, Uruguay 2: ENST Bretagne, France 3: Helsinki University of Technology, Finland

***Abstract*— In this paper we study the problem of traffic matrix estimation. The problem is ill-posed and thus some additional information has to be brought in to obtain an estimate. One common approach is to use the second moment statistics through a functional mean-variance relationship. We derive analytically the Fisher information matrix under this framework and obtain the Cramér-Rao lower bound (CRLB) for the variance of an estimator of the traffic matrix. Two applications for the use of the CRLB are then demonstrated. One is the possibility to evaluate the efficiency of an estimator against the lower bound. The other is an approach to find the best placement for direct measurements, so that it is optimal with regard to the traffic matrix estimation problem.**

I. Introduction

The traffic matrix gives the volume of traffic between each origin/destination (OD) pair in the network. While the knowledge of the traffic matrix is essential in network management and traffic engineering, it usually is not possible to measure it directly from the network. This would require Netflow or equivalent measurement devices running network-wide. This approach, however, has huge overhead because of the massive measurements, and is thus impractical in current IP networks.

The goal of traffic matrix estimation is to obtain an estimate for the traffic matrix using information which is readily available in the network: the link counts and the routing matrix.

In the network there are n OD pairs and m links. We denote the OD pair traffic volumes at time t by the n-vector $\boldsymbol{x}_t$, in which each element corresponds to an element of the traffic matrix, but the vector notation is used for computational reasons. Similarly, the link loads are denoted by m-vector $\boldsymbol{y}_t$. The $m \times n$ routing matrix is denoted $\boldsymbol{A} = (\ _{ij}) \in \quad _{m \times n}$ such that

$$_{ij} = \begin{cases} 1 & \text{if the OD pair } j \text{ is using link } i \\ 0 & \text{otherwise} \end{cases}$$

Typically, the routing matrix is assumed to be known, and we have several successive measurements of the link counts available, denoted by $\{y_1, \cdots, y_T\}$. The basic relationship between the link counts and OD counts $\boldsymbol{x}_t$ can be written as

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{x}_t, \tag{1}$$

Should we know the OD counts, it would be straightforward to calculate the link counts. However the opposite is not true, because in any realistic size network there are more links than OD pairs ($n > m$), making the problem of solving the traffic matrix from link counts and routing matrix heavily under-constrained, and thus ill-posed. To overcome this ill-posedness, some type of additional information has to be brought in. Typically either a prior obtained by the gravity method or the second moment statistics of the link counts. Reviews of proposed methods can be found e.g. in [4], [6].

More complex estimation techniques naturally often yield more accurate results. In general, we can say that there is a trade-off between the computational complexity and the accuracy of the estimate. However, no matter how elaborate the technique, there is a bound for the accuracy of the estimate. This is due to the stochastic nature of the traffic process, which makes it impossible to obtain estimate accuracy below certain level.

The traffic volume can be considered a random variable. The Fisher information matrix gives the amount of information that the observed traffic volumes carry of the underlying parameter, namely the expected traffic volume. For any unbiased estimate, the Cramér-Rao lower bound (CRLB), which is the inverse of the Fisher information matrix, gives the limit of how small variances it is possible to obtain for an estimator.

In this paper we show how to calculate the Cramér-Rao bounds for the traffic matrix estimation problem. There are many benefits of obtaining an expression for the CRLB. In synthetic data situations we can obtain sample variances of an estimator by Monte Carlo simulations, and compare to the bound in order to evaluate how close to optimality the considered method is. In real data situation, it is not possible to replicate the measurements as in synthetic simulations, but with the CRLB we can

get an idea of the uncertainty of an estimate since we know how large the variance is at least. As the Maximum likelihood estimate (MLE) is asymptotically efficient, its variance coincides with the Cramér-Rao lower bounds. Thus, we can obtain confidence intervals for the MLE directly from the bounds.

The rest of the paper is organized as follows. In section II we introduce the model and define the concepts of information matrix and missing information. In section III we derive an expression for the information matrix, which thus enables analytical expression for the CRLB. Section IV gives examples of possible applications for the result.

II. Preliminaries

In our model we assume that the OD pair traffic follows Gaussian distribution, that OD pairs are independent of each other, and also that successive measurements for each OD pair are independently and identically distributed. The expected value of OD pair counts $\boldsymbol{x}_t$ is denoted by the vector $\boldsymbol{\lambda}$ and the covariance matrix by $\boldsymbol{\Sigma}$.

$$\boldsymbol{x}_t \sim \quad (\boldsymbol{\lambda}, \boldsymbol{\Sigma}). \tag{2}$$

Furthermore, it is assumed that there is a functional relation between the mean and the variance.

$$\boldsymbol{\Sigma} = \Phi \mathrm{diag}(\boldsymbol{\lambda}^c). \tag{3}$$

This is a typical assumption in traffic matrix estimation, which enables the use of maximum likelihood approach. Through the mean-variance relation, the link covariances can be used to bring in the extra information needed to yield an estimate for the traffic matrix.

Thus, the link counts have expected value $\boldsymbol{A\lambda}$ and covariance matrix $\boldsymbol{A\Sigma A}^t$. We use the notation $\boldsymbol{\Sigma}' = \mathrm{diag}(\boldsymbol{\lambda}^c)$, so we can write

$$\boldsymbol{y}_t \sim \quad (\boldsymbol{A\lambda}, \Phi \boldsymbol{A\Sigma}' \boldsymbol{A}^t). \tag{4}$$

The parameters of the model are the n elements of $\boldsymbol{\lambda}$ and the parameters relating the mean to the variance, that is Φ and c, of which c is treated as a preset constant as in [3].

$$\boldsymbol{\Psi} = (\lambda_1, \lambda_2, \ldots, \lambda_n, \Phi). \tag{5}$$

As mentioned in the previous section, the traffic matric estimation problem is underconstrained. This makes traffic matrix estimation a missing data problem. The link counts are in this context the *incomplete data*, denoted by random vector $\boldsymbol{Y}$, while its realizations are the measurements $\boldsymbol{y}_t$ which form the *observed information*. The OD counts are the *complete data* random vector $\boldsymbol{X}$, and its values $\boldsymbol{x}_t$ would be the *complete information*. Since there are less samples in $\boldsymbol{y}_t$ than in $\boldsymbol{x}_t$, it is impossible to obtain correct values for the complete data by inference from the observed data. The difference between the two is called the *missing information*. And we can write the Missing Information Principle [2]

$$I = \ _c - I_m, \tag{6}$$

where I is the observed information, $\ _c$ is the complete information and I_m is the missing information.

The incomplete data random vector $\boldsymbol{Y}$ has probability density function (pdf) $\ (\boldsymbol{y}; \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is the vector containing the unknown parameters. The likelihood function for $\boldsymbol{\Psi}$, formed from the observed data $\boldsymbol{y}$ is

$$(\boldsymbol{\Psi}) = \ (\boldsymbol{y}; \boldsymbol{\Psi}) \tag{7}$$

The pdf of the complete data vector $\boldsymbol{X}$ is $\ _c(\boldsymbol{x}; \boldsymbol{\Psi})$, and

$$_c(\boldsymbol{\Psi}) = \ _c(\boldsymbol{x}; \boldsymbol{\Psi}) \tag{8}$$

is the complete data likelihood function that could be formed if $\boldsymbol{x}$ were observable. The log-likelihood function and the complete data log-likelihood function are denoted by

$$l(\boldsymbol{\Psi}) = \log \ (\boldsymbol{\Psi}) \qquad \textit{and} \qquad l_c(\boldsymbol{\Psi}) = \log \ _c(\boldsymbol{\Psi}) \tag{9}$$

respectively.

The gradient vector of the incomplete-data is given by

$$\boldsymbol{S}(\boldsymbol{y}; \boldsymbol{\Psi}) = \frac{l(\boldsymbol{\Psi})}{\boldsymbol{\Psi}}, \tag{10}$$

while

$$\boldsymbol{S}_c(\boldsymbol{x}; \boldsymbol{\Psi}) = \frac{l_c(\boldsymbol{\Psi})}{\boldsymbol{\Psi}} \tag{11}$$

is the gradient vector of the complete-data.

Now we can formulate the expressions for observed and expected information matrix.

Definition 1 (Observed information matrix): The incomplete-data observed information matrix is denoted by $I(\hat{\boldsymbol{\Psi}}; \boldsymbol{y})$ where

$$I(\boldsymbol{\Psi}; \boldsymbol{y}) = -\frac{^2 l(\boldsymbol{\Psi})}{\boldsymbol{\Psi} \ \ \boldsymbol{\Psi}^t}, \tag{12}$$

which is the matrix of the negative second-order partial derivatives of the log-likelihood function with respect to the elements of $\boldsymbol{\Psi}$.

Definition 2 (Expected information matrix): The incomplete-data expected (Fisher) information matrix is denoted by $\ (\boldsymbol{\Psi})$ where the maximum likelihood estimate (MLE) of $\boldsymbol{\Psi}$ is denoted by $\hat{\boldsymbol{\Psi}}$ and

$$(\boldsymbol{\Psi}) = \ _{\boldsymbol{\Psi}}(I(\boldsymbol{\Psi}; \boldsymbol{Y})) \tag{13}$$

where $\ _{\boldsymbol{\Psi}}$ denotes expectation using the parameter vector $\boldsymbol{\Psi}$.

We assume that the pdf $(\boldsymbol{y};\boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = (\ _1,\ _2,\cdots,\ _d)^t$, satisfies the regularity conditions

$$\left(\frac{l(\boldsymbol{y};\boldsymbol{\Psi})}{\boldsymbol{\Psi}}\right) = 0 \quad \forall\boldsymbol{\Psi}, \tag{14}$$

where the expectation is taken with respect to $(xy\boldsymbol{\Psi})$. These regularity condition are satisfied if it is possible to exchange the differentiation with the expectation.

Using this, we can state the following

Proposition 1: Under these regularity conditions, the expected (Fisher) information matrix $(\boldsymbol{\Psi})$ is given by

$$\begin{aligned}(\boldsymbol{\Psi}) &= \ _{\boldsymbol{\Psi}}(\boldsymbol{S}(\boldsymbol{Y};\boldsymbol{\Psi})\boldsymbol{S}^t(\boldsymbol{Y};\boldsymbol{\Psi})) \\ &= \ _{\boldsymbol{\Psi}}(I(\boldsymbol{\Psi};\boldsymbol{Y}))\end{aligned} \tag{15}$$

We close the section with one more definition.

Definition 3 (Complete-data expected information matrix): The expected information matrix corresponding to the complete data is given by

$$_c(\boldsymbol{\Psi}) = \ _{\boldsymbol{\Psi}}(I_c(\boldsymbol{\Psi};\boldsymbol{X})) \tag{16}$$

where

$$I_c(\boldsymbol{\Psi};\boldsymbol{x}) = -\frac{^2 l_c(\boldsymbol{\Psi})}{\boldsymbol{\Psi}\ \boldsymbol{\Psi}^t} \tag{17}$$

while its conditional expectation given y, is denoted by

$$_c(\boldsymbol{\Psi};\boldsymbol{y}) = \ _{\boldsymbol{\Psi}}(I_c(\boldsymbol{\Psi};\boldsymbol{X}|\boldsymbol{y}). \tag{18}$$

III. Cramér-Rao lower bound

In this section we develop an analytical expression for the Fisher information matrix, and thus for the Cramér-Rao lower bound for variance. First we consider a general multivariate Gaussian case, and then the specific problem of traffic matrix estimation with OD pairs following Gaussian distribution.

Theorem 1: Under the regularity conditions of (14), the covariance matrix of any unbiased estimator $\boldsymbol{\Psi}^*$ satisfies

$$_{\boldsymbol{\Psi}^*} - \ ^{-1}(\boldsymbol{\Psi}) \geq 0 \tag{19}$$

where "≥ 0" is interpreted as meaning that the matrix is positive semidefinite, and $^{-1}(\boldsymbol{\Psi})$ is the Fisher information matrix evaluated at the true value of $\boldsymbol{\Psi}$.

The above theorem gives the Cramér-Rao lower bound. It states that $_{\boldsymbol{\Psi}^*}$, the variance of any unbiased estimator cannot be lower than the inverse of the Fisher information matrix.

A. Information matrix for the general Gaussian case

We will set the traffic matrix framework aside for a moment and calculate the CRLB for the general Gaussian case. To avoid confusion, we use different symbols for mean and variance in this case, than those introduced for the traffic matrix estimation problem in the previous section. The incomplete data is a multivariate Gaussian with mean $\boldsymbol{\mu} = \boldsymbol{\mu}(\boldsymbol{\Psi})$ and covariance matrix $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{\Psi})$. The probability density function is

$$\begin{aligned}(\boldsymbol{y};\boldsymbol{\Psi}) &= \frac{1}{(2\ )^{m/2}\det\boldsymbol{C}(\boldsymbol{\Psi})^{1/2}}\cdot \\ \cdot\exp &\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\right. .\end{aligned} \tag{20}$$

And it follows that the log-likelihood is

$$\begin{aligned}l(\boldsymbol{y};\boldsymbol{\Psi}) &= -\log(2\ )^{m/2} - \frac{1}{2}\log\det(\boldsymbol{C}(\boldsymbol{\Psi})) \\ &-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})).\end{aligned} \tag{21}$$

An element of the information matrix can be written as

$$(\boldsymbol{\Psi})_{ij} = \left[\frac{l(\boldsymbol{y};\boldsymbol{\Psi})}{\ _i}\frac{l(\boldsymbol{y};\boldsymbol{\Psi})}{\ _j}\right] \tag{22}$$

Proposition 2: The analytical expression for the information matrix is

$$\begin{aligned}(\boldsymbol{\Psi})_{ij} &= \frac{\boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\ _i}\boldsymbol{C}^{-1}(\boldsymbol{\Psi})\frac{\boldsymbol{\mu}(\boldsymbol{\Psi})}{\ _j} \\ &+\frac{1}{2}tr\left(\boldsymbol{C}^{-1}(\boldsymbol{\Psi})\frac{\boldsymbol{C}(\boldsymbol{\Psi})}{\ _i}\boldsymbol{C}^{-1}(\boldsymbol{\Psi})\frac{\boldsymbol{C}(\boldsymbol{\Psi})}{\ _j}\right)\end{aligned} \tag{23}$$

The derivation of this expression is given in the appendix.

B. Information matrix for Gaussian origin-destination pairs

We will now return to the traffic matrix estimation problem, and use the result obtained above for the general Gaussian case. We have previously defined the vector of link counts $\boldsymbol{y}_t$ as multivariate gaussian distribution with mean

$$\boldsymbol{\mu}(\boldsymbol{\Psi}) = \boldsymbol{A}\boldsymbol{\lambda}, \tag{24}$$

and covariance matrix

$$\boldsymbol{C}(\boldsymbol{\Psi}) = \Phi\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}}, \tag{25}$$

where notation defined in section II is used. Thus the probability density function is

$$\begin{aligned}(\boldsymbol{y}_t) &= \frac{1}{(2\ )^{r/2}\det(\Phi\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{1/2}}\cdot \\ \cdot\exp &\left\{-\frac{1}{2}(\boldsymbol{y}_t-\boldsymbol{A}\boldsymbol{\lambda})^{\mathrm{t}}(\Phi\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1}(\boldsymbol{y}_t-\boldsymbol{A}\boldsymbol{\lambda})\right.\end{aligned} \tag{26}$$

Since consecutive measurement samples of the link counts are considered independent from each other, the pdf can be written in product form

$$(\boldsymbol{y}) = \prod_{t=1}^{T} (\boldsymbol{y}_t). \tag{27}$$

It follows directly that the log-likelihood can be written as a sum

$$l(\boldsymbol{y}) = \sum_{t=1}^{T} l(\boldsymbol{y}_t), \tag{28}$$

and as the information matrix is the same for each time period, beacuse of the iid property, we can write

$$(\boldsymbol{\Psi}) = T \ {}_t(\boldsymbol{\Psi}), \tag{29}$$

where

$${}_t(\boldsymbol{\Psi}) = \quad {}_{\boldsymbol{\Psi}}(\boldsymbol{S}(Y_t;\boldsymbol{\Psi})\boldsymbol{S}^{\mathrm{t}}(Y_t;\boldsymbol{\Psi})), \tag{30}$$

and $\boldsymbol{S}(Y_t;\boldsymbol{\Psi})$ is the gradient vector of the incomplete data, defined in equation (10).

An element of the information matrix for the general case was given in (23). Thus, to obtain this for the specific case in question here, we have to calculate the expressions

$$\frac{\boldsymbol{\mu}(\boldsymbol{\Psi})}{{}_i} \quad \text{and} \quad \frac{\boldsymbol{C}(\boldsymbol{\Psi})}{{}_i} \tag{31}$$

for the traffic matrix estimation problem. Inserting the expressions from (24) and (25) into (31), we obtain for $i = 1, \cdots, n$ the derivative of the mean as

$$\frac{\boldsymbol{\mu}(\boldsymbol{\Psi})}{{}_i} = \frac{\boldsymbol{A}\boldsymbol{\lambda}}{\lambda_i} = \boldsymbol{A}^i, \tag{32}$$

where $\boldsymbol{A}^i$ is the ith column of $\boldsymbol{A}$, and for the covariance matrix

$$\frac{\boldsymbol{C}(\boldsymbol{\Psi})}{{}_i} = \Phi \frac{\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}}}{\lambda_i} = \Phi \boldsymbol{A} \frac{\boldsymbol{\Sigma}'}{\lambda_i} \boldsymbol{A}^{\mathrm{t}} = \Phi c \lambda_i^{c-1} \boldsymbol{A}^i \boldsymbol{A}^{i\mathrm{t}}. \tag{33}$$

For the case $i = n+1$, the differentiation is done with regard to the parameter ${}_{n+1} = \Phi$. This yields

$$\frac{\boldsymbol{\mu}(\boldsymbol{\Psi})}{{}_i} = \frac{\boldsymbol{A}\boldsymbol{\lambda}}{\Phi} = 0 \tag{34}$$

and

$$\frac{\boldsymbol{C}(\boldsymbol{\Psi})}{{}_i} \quad = \quad \frac{(\Phi \boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})}{\Phi} = \boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}}. \tag{35}$$

The information matrix has the following form

$${}_t(\boldsymbol{\Psi}) = \left(\begin{array}{c|c} I_1 & I_2 \\ \hline I_3 & I_4 \end{array} \right), \tag{36}$$

where I_1 is a $n \times n$ matrix, I_2 is a column vector of length n, I_3 is a row vector of the same length, and I_4 is a scalar. To simplify the notation we introduce the matrix

$$\boldsymbol{W} = \boldsymbol{A}^t(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^t)^{-1}\boldsymbol{A}, \tag{37}$$

which has the elements

$$w_{ij} = \boldsymbol{A}^{j\mathrm{t}}(\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1}\boldsymbol{A}^i. \tag{38}$$

Starting from the expression obtained in (23) and using the derivatives of $\boldsymbol{\mu}(\boldsymbol{\Psi})$ and $\boldsymbol{C}(\boldsymbol{\Psi})$ derived above, we can now calculate the expressions for the elements of the information matrix. For $i, j = 1, \cdots, n$,

$$\begin{aligned}
(I_1)_{i,j} &= \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^{\mathrm{t}}}{\partial \lambda_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \lambda_j} \\
&\quad + \frac{1}{2} tr \left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \lambda_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \lambda_j} \right) \\
&= \frac{1}{\Phi} \boldsymbol{A}^{i\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^j + \frac{c^2 \lambda_i^{c-1} \lambda_j^{c-1}}{2} \cdot \\
&\quad \cdot \mathrm{tr} \left((\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^i \boldsymbol{A}^{i\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^j \boldsymbol{A}^{j\mathrm{t}} \right) \\
&= \frac{1}{\Phi} \boldsymbol{A}^{i\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^j + \frac{c^2 \lambda_i^{c-1} \lambda_j^{c-1}}{2} \cdot \\
&\quad \cdot \left(\boldsymbol{A}^{j\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^i \boldsymbol{A}^{i\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^j \right) \\
&= w_{ij} + \frac{c^2 \lambda_i^{c-1} \lambda_j^{c-1}}{2} \phi^2 w_{ij}^2.
\end{aligned} \tag{39}$$

For $i = 1, \cdots, n$

$$\begin{aligned}
&(I_2)_{i,(n+1)} \\
&= \frac{1}{2} tr \left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Phi} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \lambda_i} \right) \\
&= \frac{c\lambda_i^{c-1}}{2} tr \left(\frac{1}{\Phi} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}} \frac{1}{\Phi} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \Phi \boldsymbol{A}^i \boldsymbol{A}^{i\mathrm{t}} \right) \\
&= \frac{c\lambda_i^{c-1}}{2\Phi} tr \left((\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^i \boldsymbol{A}^{i\mathrm{t}} \right) \\
&= \frac{c\lambda_i^{c-1}}{2\Phi} \left(\boldsymbol{A}^{i\mathrm{t}} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}^i \right) \\
&= \frac{c\lambda_i^{c-1}}{2} w_{ii}
\end{aligned} \tag{40}$$

Analogously, for $j = 1, \cdots, n$

$$(I_3)_{(n+1),j} = \frac{c\lambda_j^{c-1}}{2} w_{jj} \tag{41}$$

And finally,

$$\begin{aligned}
&(I_4)_{(n+1),(n+1)} \\
&= \frac{1}{2} tr \left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Phi} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Phi} \right) \\
&= \frac{1}{2} tr \left(\frac{1}{\Phi} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}} \frac{1}{\Phi} (\boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}})^{-1} \boldsymbol{A}\boldsymbol{\Sigma}'\boldsymbol{A}^{\mathrm{t}} \right) \\
&= \frac{1}{2\Phi^2} tr \left(Id_{m \times m} \right) = \frac{m}{2\Phi^2}.
\end{aligned} \tag{42}$$

We have now obtained an analytical expression for the Fisher information matrix of the traffic matrix estimation problem. The Cramér-Rao lower bound for the variance of an estimator is then just ${}^{-1}$, where the CRLB for

variances of the parameters are the diagonal elements. In the next section we will demonstrate the benefits of obtaining the expression for the CRLB.

IV. Applications

A. Evaluation of estimation techniques

Based on synthetic data evaluation studies ([4], [5]) of traffic matrix estimation methods, it would seem that the most effective methods are the ones using the maximum likelihood approach, most notably the method by Cao et al. [3]. The problem with this approach is that even with numerical methods such as the EM algorithm [1] the method does not scale well to realistic size networks. Thus, computationally lighter approaches, such as [7], [8], that trade accuracy for computational lightness have been proposed. However, as the likelihood method scales poorly, it is difficult to make comparisons about the tradeoff between estimation accuracy and computation time in realistic size situations. Indeed, in both [7] and [8], the comparisons between the proposed methods and the full likelihood method is performed only in a small topology.

Theorem 2: [Asymptotic Efficiency of the MLE] The asymptotic covariance matrix of the MLE is equal to the inverse of the expected information matrix [1].

From simulations with synthetic traffic matrices we can obtain sample variances for the considered methods. Then calculating the CRLB, it is possible to compare them to the bound, and thus, because of Theorem 2, to the variance of MLE. This way we can evaluate how much less accurate the methods are than the full MLE, without having to run the full likelihood method.

B. Optimal location for direct measurements

Consider that we could deploy a measurement device on one link of the network. We consider here link-wise measurements, but the basic technique is the same even if measurements are available router-wise for each link adjacent to the router where the measurement device is used. For instance NetFlow [9] is capable of collecting these types of flow level measurements.

By direct OD flow measurements we can obtain the actual traffic volumes of each OD pair traversing the measured link, instead of just the total link load available by SNMP measurements, leading to more accurate estimates of the traffic matrix. To incorporate the direct measurements of some OD floes to the traffic matrix estimation framework we propose a model that creates a new linear system. This can be interpreted as a virtual topology, where the link where the direct measurements are made is replaced by several virtual links, such that each OD pair using the link would have its own virtual link. This enables us to incorporate the direct OD pair measurements without changing the basic situation. For example, The network in Figure 1 has three links (AB, BC, BD) and four OD pairs $(x_{AC}, x_{AD}, x_{BC}, x_{BD})$. The routing matrix is

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

The link B has two OD flows, x_{AD} and x_{BD}, using it. In the virtual topology the link is replaced by virtual links B_1 and B_2. Now x_{AD} is the only OD flow traversing link B_1 and x_{BD} is the only OD flow traversing link B_2. Thus the direct measurements can be incorporated into the estimate through the usual inference techniques from equation

$$\boldsymbol{y}' = \boldsymbol{A}'\boldsymbol{x},$$

where $\boldsymbol{y}'$ and $\boldsymbol{A}'$ are the link loads and the routing matrix of the virtual topology. The routing matrix changes so that the last row of A, that corresponds to link B, is divided into two rows corresponding to virtual links

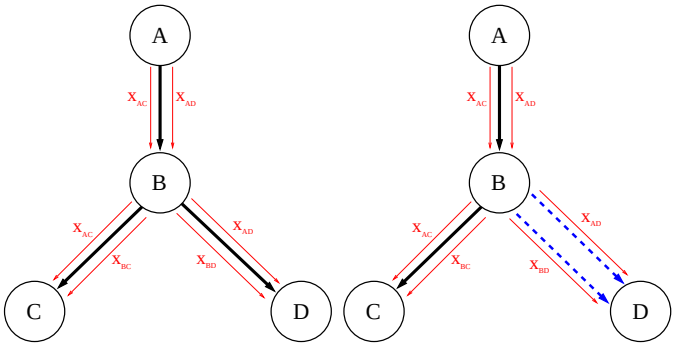


Fig. 1. Left: Example topology. Right: Link BD is replaced by virtual links BD_1 and BD_2

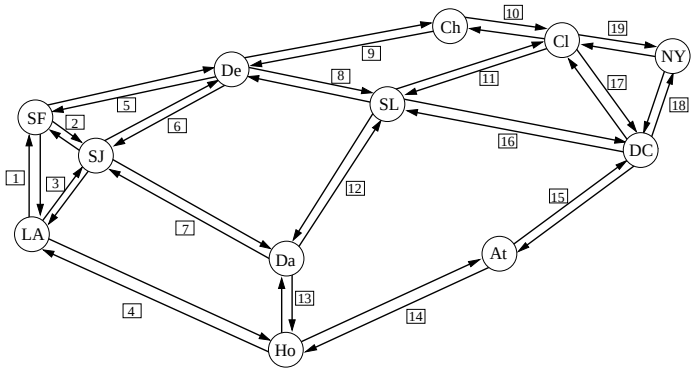


Fig. 2. Fictional US Backbone topology

TABLE I

BEST PLACEMENTS FOR A SINGLE LINK MEASUREMENT

link	18r	18	15r	9r	14	10	...	2
Avg. Var	0.58	0.72	0.72	0.73	0.73	0.74	...	0.99

TABLE II

BEST PLACEMENTS FOR TWO LINK MEASUREMENTS

link1	18r	18r	18r	18r	18r	18r	18r	18r	18r	18	18	18r
link2	18	9r	15	10	14r	19	4r	12	5r	15r	14	3r
Avg. Var	0.36	0.39	0.40	0.40	0.40	0.43	0.44	0.46	0.47	0.47	0.48	0.48

B_1 and B_2.

$$A' = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Using the equations derived earlier in this paper we can calculate the Cramér-Rao lower bounds for variances of the OD pairs for this virtual topology. If we have maximum likelihood estimates for the OD pairs, we can calculate the variance of the estimates by plugging in the MLE into the CRLB equations. As some of the OD pairs are directly observed, their variances are very small. It is not zero, however, since we are observing the OD pair traffic loads $\boldsymbol{x}$, a stochastic variable, whose expected value λ we are trying to estimate. Also the other OD pairs have now lower variances, because the new estimation problem is less underconstrained than the original.

Changing the location of the measurement device, creates different virtual topologies that lead to different OD pairs being directly observed.

Also, as this is all analytical calculations, it is rather quick to calculate all two link combinations, to find out how to best place two measurement devices.

Comparing the OD pair variances of the virtual topology to the OD pair variances of the original topology, the accuracy gained from measurements on a given link can be evaluated. Any number of criterions may be used, but for the sake of example we use the ratio of the average of the OD pair variances

$$\frac{tr(\ (\boldsymbol{A}'))/n}{tr(\ (\boldsymbol{A}))/n} \tag{43}$$

in the sequel, as this gives an good indication about how much a measurement is able to reduce the variances.

Repeating the above procedure for each link, we can compare the results each measurement would yield for the average variance, and thus be able to find the optimal location for measurement, that is, the link yielding the lowest average variance.

For example, consider the fictional US backbone topology in Figure 2, where the links have been enumerated such that we refer to the direction with the number next to it by that number, and indicate the opposite direction by lower case r. So for instance the link from LA to SF is 1 and the link in the reverse direction from SF to LA is then 1r.

We consider a traffic distribution generated by the gravity model based on the population of the cities in question. There are a few larger OD pairs, especially the ones between LA and NY, as well as from those two to the middle sized cities. The best location for the measurements is not solely dependent on the location of the link, but needs to capture as many of the bigger flows as possible. It turns out that the best placement by far is link 18r. The next best locations include the other links from the same $NY - A$ route. Table I shows the optimal links and the ratio of average variances divided by the average variances of the original case.

Also the best two link combinations are dominated by the aforementioned link 18r. Best combination is to have both links between NY and to capture the big flows between NY and A in both directions. The best placements are shown in Table II

Selecting the combination of two links from the single link calculations would in these example cases yield the optimal placements, but not always as the second best link might be capturing some of the same large OD pairs as the first link, making it good location for a single measurement point but not very reasonable for a second point if the large OD pair is already measured by the first location. This is the case, for instance, between best and third best links in the gravity case, which combination would not be very effective choice for two measurement locations. Doing the selection sequentially, on the other hand, would remove this problem and yield in most cases optimal solution reducing the running time of the

calculations from m^2 to $2m$, where m is the number of links.

V. Conclusion

In this paper we derived an analytical expression for the Fisher information matrix in the traffic matrix estimation framework. The result was used to yield the Cramér-Rao lower bound for the variance of an estimator in the situation where we assume a functional mean-variance relationship for origin-destination flows in the network. We showed how this result can be very useful in evaluation of estimation techniques, as we can compare the variance of the evaluated estimator to the lower bound to see how effective it is. Another use for the bounds is also demonstrated, where we show how to utilize the result in finding the optimal place for measurements to reduce the average error of a traffic matrix estimate as much as possible.

Appendix

A. Derivation of the general gaussian case information matrix

An element of the information matrix can be written as

$$\mathcal{I}(\boldsymbol{\Psi})_{ij} = E\left[\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_j}\right] \tag{44}$$

We will use the following identities

$$\frac{\partial \log\det(\boldsymbol{C}(\boldsymbol{\Psi}))}{\partial \Psi_i} = \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \tag{45}$$

and

$$\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} = -\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}. \tag{46}$$

Now we need to calculate the first order derivatives of the log-likelihood.

$$\begin{aligned}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i} &= -\frac{1}{2}\frac{\partial \log\det(\boldsymbol{C}(\boldsymbol{\Psi}))}{\partial \Psi_i} \\ &\quad -\frac{1}{2}\frac{\partial}{\partial \Psi_i}\left[(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\right]\end{aligned} \tag{47}$$

The first term is

$$-\frac{1}{2}\frac{\partial \log\det(\boldsymbol{C}(\boldsymbol{\Psi}))}{\partial \Psi_i} = -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right)$$

We now consider the second term:

$$\begin{aligned}&\frac{\partial}{\partial \Psi_i}\left[(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\right] \\ &\quad= \frac{\partial(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\qquad +(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\qquad +(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))}{\partial \Psi_i} \\ &\quad= -\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\qquad +(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\qquad -(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_i}\end{aligned} \tag{48}$$

Using

$$(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_i} = \left(\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\right)^t \tag{49}$$

it follows that

$$\begin{aligned}&\frac{\partial}{\partial \Psi_i}\left[(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\right] \\ &\quad= -2\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\qquad +(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\end{aligned} \tag{50}$$

Then,

$$\begin{aligned}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i} &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \\ &\quad +\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) \\ &\quad -\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))\end{aligned} \tag{51}$$

Having obtained an expression for the derivatives of the log-likelihood, we are now ready to calculate the information matrix, i.e:

$$\mathcal{I}(\boldsymbol{\Psi})_{ij} = E\left[\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_j}\right] \tag{52}$$

Let us define $\boldsymbol{z} = (\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi}))$ for a shorter notation.

$$\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_j}$$

$$= \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right)\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \tag{53}$$

$$- \frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right)\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_j}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z} \tag{54}$$

$$+ \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right)\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j}\boldsymbol{z} \tag{55}$$

$$- \frac{1}{2}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z}\,\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \tag{56}$$

$$+ \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_j}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z} \tag{57}$$

$$- \frac{1}{2}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z}\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j}\boldsymbol{z} \tag{58}$$

$$+ \frac{1}{4}\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}\boldsymbol{z}\,\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \tag{59}$$

$$- \frac{1}{2}\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}\boldsymbol{z}\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_j}\boldsymbol{C}(\boldsymbol{\Psi})^{-1}\boldsymbol{z} \tag{60}$$

$$+ \frac{1}{4}\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i}\boldsymbol{z}\boldsymbol{z}^t\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j}\boldsymbol{z} \tag{61}$$

Since

$$E(\boldsymbol{z}) = E(\boldsymbol{y}-\boldsymbol{\mu}(\boldsymbol{\Psi})) = 0 \tag{62}$$

and all odd order moments are also zero, terms (54),(56),(58) and (60) are zero in the above equation. For calculating the expectation

$$E\left(\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i}\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_j}\right) \tag{63}$$

we need to consider the expectations of terms (53), (55), (57), (59) and (61).

Before we compute the expected value of (55) let us first compute

$$\left(\ ^t \frac{(\Phi)^{-1}}{\Phi_j} \ \right).$$

Since

$$\left(\ ^t \ \right) = \mathrm{tr}\left(\quad \left(\quad ^t \right) \right)$$

we have that

$$\begin{aligned} E\left(\boldsymbol{z}^t \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{z}\right) &= \mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} E\left(\boldsymbol{z}\boldsymbol{z}^t\right)\right) \\ &= \mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})\right) \\ &= -\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right), \end{aligned} \quad (64)$$

where the least equality follows from using equation (46). So the expected value of (55) is

$$\begin{aligned} &\frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) E\left(\boldsymbol{z}^t \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{z}\right) \\ &\quad = -\frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \end{aligned} \quad (65)$$

Analogously for (59),

$$\begin{aligned} &E\left(\frac{1}{4}\boldsymbol{z}^t \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{z}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\quad = -\frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \end{aligned} \quad (66)$$

Let us now compute the expected value of (57).

$$\begin{aligned} &E\left(\frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \boldsymbol{z} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \boldsymbol{z}\right) \\ &\quad = \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} E\left(\boldsymbol{z}\boldsymbol{z}^t\right) \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_j} \\ &\quad = \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_j} \end{aligned} \quad (67)$$

Before we compute the expected value of (61) let us remind that if $_1$ and $_2$ are symmetric then

$$E(y^t D_1 y y^t D_2 y) = \mathrm{tr}(D_1 D)\mathrm{tr}(D_2 D) + 2\mathrm{tr}(D_1 D D_2 D)$$

where $= (yy^t)$. Using that result it comes that the expected value of (61) is

$$\begin{aligned} &E\left(\frac{1}{4}\boldsymbol{z}^t \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{z}\boldsymbol{z}^t \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{z}\right) \\ &\quad = \frac{1}{4}\mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi})\right) \mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})\right) \\ &\qquad + \frac{1}{2}\mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})\right) \\ &\quad = \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\qquad + \frac{1}{2}\mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})\right), \end{aligned} \quad (68)$$

where the final equality follows from using (46) on the first term.

Thus, noting that the expectation of (53) is just the term itself and taking the terms (65),(67),(66), and (68), we get

$$\begin{aligned} &E\left(\frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_i} \frac{\partial l(\boldsymbol{y};\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\quad = \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\qquad - \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\qquad + \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_j} \\ &\qquad - \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\qquad + \frac{1}{4}\mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i}\right) \mathrm{tr}\left(\boldsymbol{C}(\boldsymbol{\Psi})^{-1} \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \\ &\qquad + \frac{1}{2}\mathrm{tr}\left(\frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_i} \boldsymbol{C}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})^{-1}}{\partial \Psi_j} \boldsymbol{C}(\boldsymbol{\Psi})\right). \end{aligned} \quad (69)$$

And finally, as the first and second, as well as the fourth and fifth term above cancel each other out, we get

$$\begin{aligned} \mathcal{I}(\boldsymbol{\Psi})_{ij} &= \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})^t}{\partial \Psi_i} \boldsymbol{C}^{-1}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{\mu}(\boldsymbol{\Psi})}{\partial \Psi_j} \\ &\quad + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{C}^{-1}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_i} \boldsymbol{C}^{-1}(\boldsymbol{\Psi}) \frac{\partial \boldsymbol{C}(\boldsymbol{\Psi})}{\partial \Psi_j}\right) \end{aligned} \quad (70)$$

REFERENCES

[1] G.J. McLachlan and T. Krishnan, "The EM Algorithm and Extensions", *John Wiley and Sons, Inc*, 1997.

[2] T. Orchard and M.A. Woodbury, "A missing information principle: theory and applications". Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability, vol **1**. Berkeley, California: University of California Press, pp. 697-715, 1972.

[3] J. Cao, D. Davis, S. V. Wiel, B. Yu, "Time-varying network tomography," Journal of the American Statistical Association, vol. 95, pp. 1063–1075, 2000.

[4] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions", ACM SIGCOMM, Pittsburg, USA. August 2002.

[5] A. Medina, K. Salamatian, N. Taft, I. Matta, Y. Tsang, C. Diot, "On the Convergence of Statistical Techniques for Inferring Network Traffic Demands", Tecnhical Report, 2003.

[6] S. Vaton, J.S. Bedo, A. Gravey, "Advanced methods for the estimation of the Origin Destination traffic matrix", Revue du 25ème anniversaire du GERAD, 2005.

[7] J. Cao, S. V. Wiel, B. Yu, Z. Zhu, "A scalable method for estimating network traffic matrices," Bell Labs Technical Report, 2001.

[8] G. Liang, B. Yu, "Pseudo Likelihood Estimation in Network Tomography", in Infocom 2003.

[9] Cisco NetFlow, www.cisco.com/warp/public/732/Tech/netflow.

Computer Networks 54 (2010) 1750–1766

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Optimal volume anomaly detection and isolation in large-scale IP networks using coarse-grained measurements

P. Casas [a,c,*], S. Vaton [a], L. Fillatre [b], I. Nikiforov [b]

[a] Télécom Bretagne, Brest, France
[b] Université de Technolgie de Troyes, Troyes, France
[c] Universidad de la República, Montevideo, Uruguay

ARTICLE INFO

Article history:
Received 10 August 2009
Received in revised form 13 January 2010
Accepted 23 January 2010
Available online 4 February 2010
Responsible Editor: A. Popescu

Keywords:
Network Monitoring and Traffic Analysis
Traffic Matrix
Network Traffic Modeling
Optimal Volume Anomaly Detection and Isolation

ABSTRACT

Recent studies from major network technology vendors forecast the advent of the Exabyte era, a massive increase in network traffic driven by high-definition video and high-speed access technology penetration. One of the most formidable difficulties that this forthcoming scenario poses for the Internet is congestion problems due to traffic volume anomalies at the core network. In the light of this challenging near future, we develop in this work different network-wide anomaly detection and isolation algorithms to deal with volume anomalies in large-scale network traffic flows, using coarse-grained measurements as a practical constraint. These algorithms present well-established optimality properties in terms of false alarm and miss detection rate, or in terms of detection/isolation delay and false detection/isolation rate, a feature absent in previous works. This represents a paramount advantage with respect to current in-house methods, as it allows to generalize results independently of particular evaluations. The detection and isolation algorithms are based on a novel linear, parsimonious, and non-data-driven spatial model for a large-scale network traffic matrix. This model allows detecting and isolating anomalies in the Origin-Destination traffic flows from aggregated measurements, reducing the overhead and avoiding the challenges of direct flow measurement. Our proposals are analyzed and validated using real traffic and network topologies from three different large-scale IP backbone networks.

1. Introduction

After a brief mid-decade slowdown, IP traffic will nearly double every 2 years in the near future. The overall IP traffic is expected to grow from 6.6 exabytes per month in 2007 to nearly 29 exabytes per month by 2011 (1 exabyte = 10^{18} bytes), more than quadrupling in less than a half decade [1]. Simultaneously, the evolution of access technologies and the development of optical access networks (Fiber To The Home technology) will dramatically increase the bandwidth for end-users, imposing serious and unforeseen problems at the core network, so far assumed infinitely provisioned. One of the most difficult challenges for network operators will be to correctly manage the large and unexpected congestion problems at the core network caused by volume anomalies. These observations are part of the key findings provided by Cisco's global IP traffic forecast 2006-2011 [1,2].

Volume anomalies represent large and sudden link load changes due to strong variations in traffic flows. These variations arise from unexpected events such as flash crowds, network equipment failures, network attacks, and external routing modifications and traffic shifts. Large-scale monitoring systems are currently deployed in

* Corresponding author. Address: Université de Technolgie de Troyes, Troyes, France.
E-mail addresses: pedro.casas@telecom-bretagne.eu (P. Casas), sandrine.vaton@telecom-bretagne.eu (S. Vaton), lionel.fillatre@utt.fr (L. Fillatre), igor.nikiforov@utt.fr (I. Nikiforov).

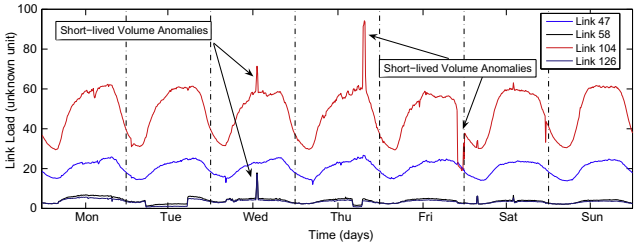
1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.01.013

ISP (Internet Service Providers) and large enterprise networks to fight back against these unexpected events. In this work we focus on two central aspects of traffic monitoring for volume anomaly detection: (i) the rapid and accurate detection of volume anomalies and (ii) the isolation of the origins of the detected anomalies.

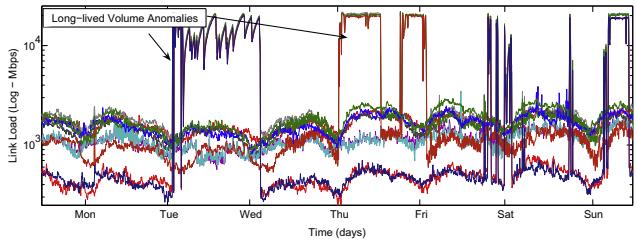
The first issue corresponds to the anomaly detection field, a difficult and extensively studied problem. Anomaly detection in data networks consists of identifying patterns that deviate from normal traffic behavior. Different types of network anomalies can be detected depending on the monitored data. We particularly focus on device-level data. Device monitoring consists in using the widely spread Simple Network Management Protocol (SNMP) to periodically collect management device readings, known as Management Information Base (MIB) variables. Every network device has a set of MIB variables that are specific to its functionality, like memory usage, CPU load, and interface bandwidth usage among others. SNMP is unique in that it is supported by basically every device in an IP network. SNMP is the most basic means of data collection for traffic analysis and provides the most coarse-grained information. At the same time it is the technique that causes the least measuring overhead, and thus represents an appealing choice for large-scale monitoring. However, it also has practical limitations, like missing data due to the use of the unreliable UDP transport protocol to export readings, or lack of readings synchronization in large-scale networks.

In this work we focus on network-wide volume anomaly detection, analyzing network traffic at the Origin-Destination (OD) flow level. An OD flow represents the total aggregated traffic flows transmitted between an ingress and an egress node or PoP (Point of Presence) in a network. A network-wide view of OD flows within a network is typically described by a Traffic Matrix (TM); a TM represents the total volume of traffic transmitted between every pair of ingress and egress points of a network. In practice, the term "volume of traffic" refers to the cumulative number of bytes between two consecutive measurements. The TM is a volume representation of OD flows traffic, and thus the types of anomalies we can expect to detect from its analysis are volume anomalies. Fig. 1 depicts the occurrence of short-lived (a couple of hours at most) and long-lived volume anomalies in 1(a) four monitored links from a commercial international Tier-2 network and 1(b) several links from the Abilene network, an Internet2 backbone network in the US. As each OD flow typically spans multiple network links, a volume anomaly in one single OD flow is simultaneously visible on several links.

The algorithms that we develop in this work make use of standard SNMP per-link byte counts to detect volume anomalies in the TM. Link byte counts represent the accumulated number of bytes that cross through the link between two consecutive readings. From now on, we shall use the term "SNMP measurements" as a reference to this link data. The use of this aggregated coarse-grained data allows to conceive light and easy-to-deploy anomaly detection/isolation algorithms. However, it poses a challenging problem: the number of links in a network is generally much smaller than the number of OD flows, and thus the TM is not directly observable from link measurements.



(a) One week of traffic in a Tier-2 ISP network, corresponding to 1008 consecutive measurements.



(b) One week of traffic in the Abilene network, corresponding to 2016 consecutive measurements.

Fig. 1. Network volume anomalies in large-scale IP networks. Each measurement corresponds to the cumulative number of bytes between two consecutive SNMP readings.

The second issue that we address is the isolation of the origins of a detected anomaly. The isolation of an anomaly consists in inferring the exact location of the problem from a set of observed anomaly indications. This represents another critical task in network monitoring, given that a correct isolation may represent the difference between a successful or a failed countermeasure. In this work we assume that traffic anomalies are exogenous unexpected events (flash crowds, external routing modifications, external network attacks) that *significantly* modify the volume of one or multiple OD flows within the monitored network. For this reason, the isolation of the anomaly consists in finding the OD flows that suffer such a variation, referred from now on as the *anomalous* OD flows.

1.1. Related work

The anomaly detection literature treats the detection of general anomalous traffic behaviors [14,25,26,28,31,33] as well as specific kinds of network and traffic anomalies. A basic list includes flash crowd events [12,13], network failures [9–11,15,30], network attacks [16,18,19,32], and large traffic shifts [20] among others. The majority of these works operate on individual and independent time series, analyzing traffic at a particular network link, particular device readings or particular packet characteristics with classical forecasting and outliers analysis methods. For example, [15] uses exponential smoothing EWMA and Holt-Winters forecasting techniques to detect anomalous behaviors in router readings. [14] analyses frequency characteristics of flow traffic and SNMP measurements using a wavelets based filtering approach, exposing anomalies as sharp variations in the filtered data variance. [17] builds compact summaries of flow traffic data using the notion of *sketch*, applying then the same forecasting techniques used in previous works (ARIMA, Holt-Winters, etc.) on

top of such summaries to detect significant forecast errors. [16] uses spectral analysis techniques over TCP flows for DoS (Denial of Service) detection, using traffic traces from a single network link. [20] uses BGP (Border Gateway Protocol) and SNMP data streams to detect large traffic shifts, using EWMA, seasonal analysis and Holt-Winters over single time series to filter periodic and trend components, detecting anomalies as impulse functions. [12] characterizes flash crowds in Web servers and provides a network aware clustering approach to distinguish these events from DoS attacks, proposing an adaptive CDN (Content Delivery Network) architecture to fight back against these extreme events. [30] represents one of the first papers that uses multiple time series for anomaly detection, synthesizing information from multiple MIB variables at a single router to improve results. Contrary to these works, we treat the anomaly detection problem from a network-wide perspective, exploiting spatial correlations across the time series of traffic from all the links of a network.

Network-wide anomaly detection has also been treated in different works [25–28,31,33]. The methods proposed in [26–28] make use of rich flow and packet data to detect anomalies, but this data that can be too costly to collect and to process [3]. [25] detects and classifies anomalies by jointly analyzing the distribution of OD flows and traffic features (IP addresses and ports). The authors use the technique of Principal Components Analysis (PCA) and the subspace method previously introduced in the field of fault diagnosis for chemical engineering processes [21,22] to analyze the ensemble of OD flows and the corresponding traffic features in a network. [27] uses the idea of sketch proposed in [17] and the PCA approach to identify anomalous traffic flows. [28] proposes a recursive method to detect anomalies in multivariate time series, which is validated using the number of packets and the number of individual IP flows aggregated in a TM. On the contrary, our methods make use of easy to collect coarse-grained SNMP link data to detect and isolate volume anomalies in OD flows.

The use of SNMP measurements to detect volume anomalies in OD flows has been considered in [25,31,33], but none of these works has provided a complete and reliable solution to the problem. [31] uses a Kalman-filtering approach to track the evolution of OD flows from SNMP measurements, detecting anomalies as large prediction errors. The method requires a long training phase where direct anomaly-free OD flow measurements are used to calibrate the underlying model. As we have recently shown [8], the assumed model has a particular structure that may require several periodical recalibrations to provide reliable results, which makes the method too costly to implement from a practical point of view. Besides, the paper does not tackle the anomaly isolation problem. Only [25,33] treat the problem of both anomaly detection and isolation in OD flows from SNMP measurements. The authors of [25] use the PCA approach and the subspace method proposed in [21,22] to separate SNMP measurements into a normal subspace and an anomalous subspace, where anomalies are detected. The use of the PCA technique and the subspace method has probably become the most famous approach for network-wide anomaly detection in recent years. However, the approach is a pure data-driven in-house method, and recent works [28,29] have shown categorical evidence about its serious shortcomings for anomaly detection and isolation in data networks. Finally, our approach falls into the same category as [33], where anomalies are inferred from aggregated data by combining network tomography and anomaly detection techniques. [33] uses similar methods applied in previous works to detect volume anomalies in OD flows: Fourier and Wavelet analysis, ARIMA modeling and PCA decomposition. The isolation of anomalies is performed with different heuristics which are not evaluated from a complexity perspective and that might be too time-consuming for on-line application; in fact, all evaluations performed in [33] are conducted off-line over individual datasets spanning 1 week of traffic each. Unlike that work, we provide detection and isolation algorithms that can be applied in an on-line fashion with solid theoretical support on their optimality properties.

1.2. Contributions of the paper

Despite the large literature in the field, we can see that to date there is no single approach to correctly detect and isolate traffic anomalies in the TM from SNMP measurements in an on-line fashion. A reliable implementation of such approach would be highly beneficial for network operators, providing a light and easy to deploy first-line monitoring tool for on-line anomaly detection and isolation.

In this paper we present a complete approach that meets these criteria with solid optimality properties in terms of false alarm and miss detection rate, or in terms of detection/isolation delay and false detection/isolation rate, a feature absent in previous works. Optimality support is fundamental in the conception of general algorithms, not tied to any particular network and more important, independent of particular evaluations in particular network and traffic scenarios. In-house methods may work rather well in certain scenarios, but without a principled and generalizable support they can be easily rebutted.

We begin by introducing a new parsimonious, linear and parametric model for the anomaly-free TM. This model presents important advantages: (i) it uses exclusively easily-available coarse-grained SNMP measurements, simplifying practical issues; (ii) it is non-data-driven and as we will show through evaluation with real data, it is stable in time, making it possible to design reliable anomaly detection methods on top of it; (iii) it is easy to calibrate and needs a very small amount of anomaly-free data to provide solid results; (iv) using this parsimonious model we can remove the anomaly-free traffic from the anomaly detection problem, thus treating the detection and isolation of volume anomalies as a sequential change detection/isolation problem with a nuisance parameter. This problem has been previously studied with some significant results [39–41]. In our particular case, this allows to design optimal algorithms for volume anomaly detection and isolation, using the principles of the decision theory.

Based on this traffic model, we propose two different optimal algorithms for volume anomaly detection and iso-

lation. The first algorithm is designed for optimal detection, maximizing the correct detection for a bounded false alarm rate. The second algorithm permits to simultaneously detect and isolate a particular anomalous OD flow within the TM, minimizing the maximum mean detection/isolation delay for given bounds in the false isolation and false alarm rates. Since a few anomaly-free SNMP measurements are sufficient to obtain a reliable model for the anomaly-free TM, we claim that the proposed methods are well adapted to dynamic routing scenarios and non-stationary traffic, but this case is out of the scope of the current study.

To provide strong evidence on the effectiveness of our methods, all the proposed algorithms are validated using real traffic data from three different backbone networks: the Internet2 Abilene backbone network, the European GEANT academic network, and a commercial international Tier-2 network. Additionally, we compare our algorithms against well-known works in the field, showing that similar or even better performance can be achieved with thorough theoretical foundation. This work represents a continuation of our previous works on traffic modeling and volume anomaly detection [4].

The remainder of this paper is organized as follows. In Section 2 we present a linear parsimonious model to describe the anomaly-free OD flows traffic. Section 3 presents an optimal volume anomaly detection algorithm that maximizes the power of the test for a given false alarm rate. Section 4 presents a recursive algorithm for simultaneously detecting and isolating volume anomalies in single OD flows, minimizing the maximum mean detection/isolation delay for bounded false alarm and false isolation rates. In Section 5 we present an in-depth validation and evaluation of the traffic model and the detection/isolation algorithms, comparing their performance against well-known algorithms previously proposed. Section 6 discusses complexity and implementation issues of the proposed algorithms. Finally, Section 7 concludes this work.

2. Linear parsimonious traffic matrix modeling

The first and maybe the most critical step in anomaly detection is to conceive an accurate and stable traffic model for what constitutes an anomaly-free behavior. In this work we intend to detect volume anomalies in a backbone TM from SNMP measurements, thus we develop a traffic model for the anomaly-free behavior of the OD flows within a large-scale IP network. Throughout the paper, the vector $X_t = \{x_t(1), \ldots, x_t(m)\}^T$ represents the value of the TM at time t, where $x_t(k)$ stands for the traffic volume of each OD flow $k = 1, \ldots, m$ at measurement time t. Similarly, the SNMP measurements vector $Y_t = \{y_t(1), \ldots, y_t(r)\}^T$ represents the links traffic volume at time t, where $y_t(i)$ represents the total traffic volume in link $i = 1, \ldots, r$ at measurement time t. The TM X_t and the links traffic Y_t are related through the routing matrix R:

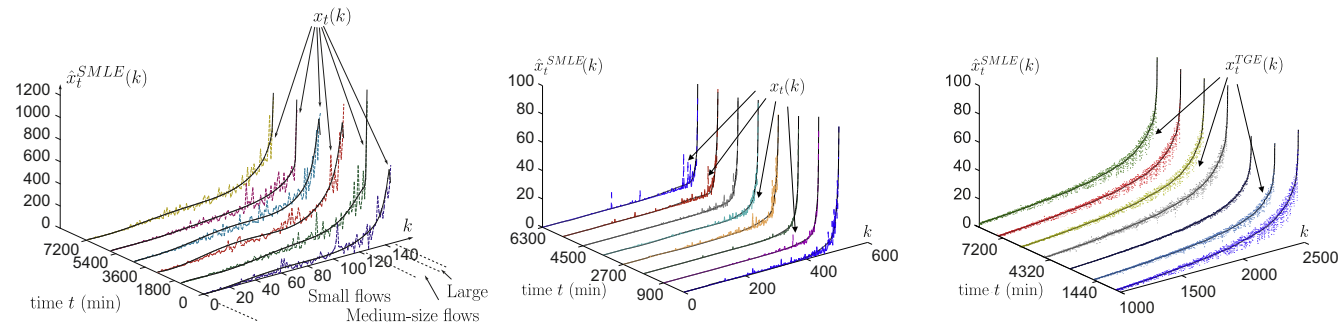
$$Y_t = RX_t \tag{1}$$

where R_{ij} is equal to 1 if OD flow j traverses link i and 0 otherwise. Note that we have intentionally omitted the subscript t in the routing matrix R; in this work we assume that R is constant in time. In Section 6 we discuss the implications of this choice.

Monitoring the behavior of X_t based on R and Y_t data represents a poorly posed problem, because the number of unknown OD flows is much larger than the number of links, $m \gg r$. To solve this problem, we propose a spatial, linear, and low-dimensionality representation of X_t in the absence of volume anomalies. The basic idea of this model is that the traffic flows X_t, sorted from smallest to largest traffic volume can be decomposed at every time t over a known family of q basis functions (columns of the matrix S), $S = \{\mathbf{s}(1), \mathbf{s}(2), \ldots, \mathbf{s}(q)\}$, with the great virtue that $q \ll m$, even several orders of magnitude smaller (in the evaluation we show that $q < 10$ even for a network with more than $m > 1000$ OD flows). Therefore, we assume that X_t can be expressed as:

$$X_t = S\boldsymbol{\mu}_t + \boldsymbol{\xi}_t \tag{2}$$

where $\boldsymbol{\xi}_t$ is a white Gaussian noise with covariance matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_m^2)$ that models the natural variability of the OD flows together with the modeling errors. The vector $\boldsymbol{\mu}_t = \{\mu_t(1) \ldots \mu_t(q)\}^T$ is an unknown time-varying vector which describes the OD flows intensity distribution with respect to the set of vectors $\mathbf{s}(i)$. We found in [8] that the order of OD flows sorted from smallest to largest traffic volume remains reasonably stable in time for several days in various large-scale networks, different not only in the



(a) Abilene Network (b) GEANT Network (c) Tier-2 ISP Network

Fig. 2. Approximation of real OD flows (dashed lines) by the spline-based (SB) model (full lines) in 3 operational networks. $x_t(k)$ is the real volume of OD flow k. $\hat{x}_t^{SMLE}(k)$ stands for the estimated OD flow k using the SB model, defined in Eq. (6). $x_t^{TGE}(k)$ is the estimated OD flow k using the tomogravity estimation method, introduced in [5].

topology but also in the nature of traffic. Fig. 2 shows the OD flows traffic for (a) the Abilene network, (b) the GEANT network, and (c) a commercial Tier-2 ISP network, sorted from smallest to largest traffic volume, for different times t.

The sorted volumes of OD flows can be approximated by a non-decreasing function with a certain smoothness. The curve obtained by interpolating this function is parameterized by using a polynomial splines approximation. Given the shape of this curve, a cubic splines approximation is used. The spline basis is finally designed to approximate the sorted volume of OD flows by using m points uniformly chosen in the interval $[1;m]$. The vectors $\mathbf{s}(i)$ in S form the set of basis vectors that describe the spatial distribution of the traffic. From now on, we shall refer to this Spline-Based model as the SB model.

To illustrate the structure of the matrix S, let us consider the polynomial splines of degree $p=3$ with $p-1$ continuous derivatives and two integer knots k_1 and k_2 such that $1<k_1<k_2<m$. A natural cubic spline $c(x)$ with the two knots k_1 and k_2 has the form:

$$c(x)=\mu(1)+\mu(2)x+\mu(3)x^2+\mu(4)x^3+\mu(5)(x-k_1)_+^3 + \mu(6)(x-k_2)_+^3$$

where x belongs to a real interval $[a;b]$ containing $[1;m]$, i.e. $[1;m]\subseteq[a;b]$, the reals $\mu(i)$ are the spline coefficients a $(x)_+=\max\{0,x\}$. The interested reader can find additional information on splines representations in [34]. Then, the sampled vector $\mathbf{c}=(c(k))_{1\leqslant k\leqslant m}$ verifies $\mathbf{c}=V\boldsymbol{\mu}$ where the matrix V is given by:

$$V=\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0\\ 1 & 2 & 2^2 & 2^3 & 0 & 0\\ 1 & \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & \vdots & \vdots & \vdots & 0 & 0\\ 1 & \vdots & \vdots & \vdots & 1 & 0\\ 1 & \vdots & \vdots & \vdots & 2^3 & 0\\ 1 & \vdots & \vdots & \vdots & \vdots & 1\\ 1 & \vdots & \vdots & \vdots & \vdots & 2^3\\ 1 & \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & m & m^2 & m^3 & (m-k_1)^3 & (m-k_2)^3 \end{pmatrix}$$

The matrix S is obtained from V by permuting the rows according to the OD flows sorting order: the ith row of S is the jth row of V, provided that the OD flow i becomes the jth OD flow after sorting from smallest to largest OD traffic volume.

It should be clear to the reader that the SB model cannot be generalized to all network topologies and scenarios, but that it holds for networks with large traffic aggregation. In the evaluation we show that this model provides accurate results for different network topologies and traffic scenarios, including a commercial network, the GEANT academic network, and Abilene, a network topology/traffic that is usually used as benchmark regarding TM studies. The dashed lines in Fig. 2 depict the value of each sorted OD flow $x_t(k), k=1\ldots m$, the full lines represent the splines approximation of the sorted flows. In order to appreciate the time stability of this approximation, the curves are plotted for various consecutive days. From (1) and (2), we can express links traffic as a function of $\boldsymbol{\mu}_t$:

$$Y_t=G\boldsymbol{\mu}_t+\boldsymbol{\upsilon}_t, \tag{3}$$

where $G=RS$ and $\boldsymbol{\upsilon}_t\sim\mathcal{N}(0,\Phi)$, with $\Phi=R\Sigma R^T$. It is assumed that G is a full column rank matrix. In fact, since the number of columns in G is very small, the product RS and its rank can be computed very fast. To simplify notation and computations, we introduce the standardized measurements vector Z_t:

$$Z_t=\Phi^{-\frac{1}{2}}Y_t=H\boldsymbol{\mu}_t+\boldsymbol{\delta}_t, \tag{4}$$

where $H=\Phi^{-\frac{1}{2}}G$, $\boldsymbol{\delta}_t\sim\mathcal{N}(0,I_r)$ and I_r is the $r\times r$ identity matrix. The purpose of this transformation is simply to reduce a given noise covariance matrix to the identity one.

If the covariance matrix Σ is unknown some additional experiments should be done. The solution consists in computing an empirical covariance matrix $\widehat{\Sigma}$ from a few measurements; in Section 5.2 we show that using just 1 h of SNMP measurements is enough to provide proper results. Some very basic results on the estimation of $\widehat{\Sigma}$ can be found in [35].

In this work we use this low-dimensionality model to filter the contribution of the anomaly-free traffic into the SNMP measurements, producing residuals sensitive to volume anomalies. As we explain in the following section, we treat the detection of volume anomalies as a statistical decision problem with a nuisance parameter, represented by the anomaly-free traffic. This allows to infer anomalies in the TM directly from aggregated data, without the preliminary TM estimation step. This approach clearly improves the accuracy and reduces the detection delay, because it does not drag possible errors from previous steps. Nevertheless and in order to validate the SB traffic model, we will use it to infer a TM from SNMP measurements in the validation Section 5.2.

The TM can be easily inferred from SNMP measurements using Eq. (4). We particularly use a maximum likelihood estimation approach to compute an estimated traffic matrix. The maximum likelihood estimate presents well-established statistical properties [35]. Since the traffic linear model (4) is Gaussian, the maximum likelihood estimate of $\boldsymbol{\mu}_t$, namely $\hat{\boldsymbol{\mu}}_t^{MLE}$ corresponds to the least mean squares estimation:

$$\hat{\boldsymbol{\mu}}_t^{MLE}=(H^TH)^{-1}H^TZ_t \tag{5}$$

This finally leads to the estimate of the traffic matrix X_t, which we will refer as the Spline-based Maximum Likelihood Estimate (SMLE) $\widehat{X}_t^{SMLE}$, defined by:

$$\widehat{X}_t^{SMLE}=S\hat{\boldsymbol{\mu}}_t^{ML}=\left(S(H^TH)^{-1}H^T\Phi^{-\frac{1}{2}}\right)Y_t \tag{6}$$

3. Optimal volume anomaly detection

The goal of the proposed detection algorithm is to detect the presence of an additive anomaly $\boldsymbol{\varphi}$ in one or more OD flows of the traffic matrix X_t from the SNMP measurements vector Y_t, with the highest probability of detection

for a given upper bounded probability of false alarm. The detection of this anomalous variation can be treated as a hypothesis testing problem, considering two possible traffic situations or hypotheses: the null hypothesis $\mathscr{H}_0$, where OD flows are anomaly-free, and the alternative hypothesis $\mathscr{H}_1$, where OD flows present an anomaly and thus traffic is no longer characterized by our anomaly-free traffic model (3). For every new SNMP measurement, the method has to choose between $\mathscr{H}_0$ and $\mathscr{H}_1$ with the "best detection performance". We shall explain below what do we mean by best detection performance.

In order to continuously adapt the decision thresholds of the method, the previously introduced anomaly-free traffic model is slightly modified, explicitly considering the temporal variation of the covariance matrix Σ. The Gaussian noise ξ_t is now assumed to have a covariance matrix $\gamma_t^2\Sigma$; the matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_m^2)$ is assumed to be known and stable in time. The scalar γ_t is unknown and serves to model the mean level of OD flows volume variance.

Considering Eq. (4), the previous hypothesis testing problem can be formulated as follows:

$$\mathscr{H}_0 = \{Z \sim \mathscr{N}(\boldsymbol{\varphi} + H\boldsymbol{\mu}, \gamma_t^2 I_r);\ \boldsymbol{\varphi} = 0,\ \boldsymbol{\mu} \in \mathbb{R}^q\} \tag{7}$$

$$\mathscr{H}_1 = \{Z \sim \mathscr{N}(\boldsymbol{\varphi} + H\boldsymbol{\mu}, \gamma_t^2 I_r);\ \boldsymbol{\varphi} \neq 0,\ \boldsymbol{\mu} \in \mathbb{R}^q\} \tag{8}$$

where $\boldsymbol{\varphi}$ represents an anomaly. Note that we have intentionally removed the time index t from Z and $\boldsymbol{\mu}$, explicitly stating that the test is applied for a single measurements vector $Z = Z_t$ at a certain time t. In the anomaly detection problem, the modeled anomaly-free traffic $\boldsymbol{\mu}$ is considered as a nuisance parameter since (i) it is completely unknown, (ii) it is not necessary for the detection and (iii) it could possibly mask the anomalies. In order to remove the nuisance parameter from the detection problem, the standardized measurements vector Z is projected onto the left null space of H, using the projection matrix $P_H^{\perp} = I_r - H(H^TH)^{-1}H^T$. Briefly speaking, we remove the "interference" of $\boldsymbol{\mu}$ from the problem. For this reason it is possible to chose between $\mathscr{H}_0$ and $\mathscr{H}_1$, provided that the projection of the anomaly $\boldsymbol{\varphi}$ onto the left null space of H is nonzero. For example, suppose that a volume anomaly of size θ occurs in OD flows j and k; then it is easy to see that $\boldsymbol{\varphi} = \theta\Phi^{-\frac{1}{2}}\mathbf{r}$, where $\mathbf{r}$ stands for the sum of the normalized columns $\mathbf{r}_j$ and $\mathbf{r}_k$ of the routing matrix R.

The quality of a statistical test is defined by the false alarm rate and the power function. The above mentioned testing problem is difficult because (i) $\mathscr{H}_0$ and $\mathscr{H}_1$ are composite hypotheses and (ii) there is an unknown nuisance parameter $\boldsymbol{\mu}$. A composite hypothesis refers to a statistical hypothesis that does not completely specify the probability distribution of the test statistic, i.e. it does not reduce to a single point into the probability space. There is no general way to test between composite hypotheses with a nuisance parameter.

Let K_α be the class of tests with an upper bounded maximum false alarm probability, $K_\alpha = \{\phi : \sup_\mu \Pr_{\varphi=0,\mu}(\phi(Z) = H_1) \leqslant \alpha\}$, $0 < \alpha < 1$. The probability $\Pr_{\varphi=0,\mu}$ stands for the measurements vector Z being generated by the distribution $\mathscr{N}(H\boldsymbol{\mu}, \gamma_t^2 I_r)$, and α is the prescribed upper bound for the probability of false alarm. The power function or hit rate is defined by the probability of correct detection $\beta_\phi(\boldsymbol{\varphi}, \boldsymbol{\mu}) = \Pr_{\varphi\neq 0,\mu}(\phi(Z) = \mathscr{H}_1)$. A priori, the power function depends on the parameter $\boldsymbol{\varphi}$ as well as on the nuisance parameter $\boldsymbol{\mu}$, which is highly undesirable.

In this work we use the statistical test $\phi^* : \mathbb{R}^r \mapsto \{\mathscr{H}_0, \mathscr{H}_1\}$ of [37,38], inspired by the fundamental paper of Wald [36]. To solve this problem, Wald [36] proposes a test $\phi^*(\cdot) \in K_\alpha$, which has uniformly best constant power (UBCP) in the class K_α over a certain family of surfaces S. The adaptation of Wald's theory to the problem with nuisance parameters in the case of problem (7) and (8) has been done in [37,38] by using the theory of invariant tests. Here, the family of surfaces of constant power $S = \{S_c : c \geqslant 0\}$ is defined by $S_c = \{\boldsymbol{\varphi} : \|P_H^{\perp}\boldsymbol{\varphi}\|^2 = c^2\}$. The UBCP invariant test realizes the best possible constant power $\beta_{\phi^*}(\boldsymbol{\varphi}, \boldsymbol{\mu}) = \beta_{\phi^*}(\boldsymbol{\varphi}', \boldsymbol{\mu}), \forall \boldsymbol{\varphi}, \boldsymbol{\varphi}' \in S_c$ and $\beta_{\phi^*}(\boldsymbol{\varphi}, \boldsymbol{\mu}) \geqslant \beta_\phi(\boldsymbol{\varphi}, \boldsymbol{\mu})$ over the tests with a given false alarm rate $\phi \in K_\alpha$. Finally, the threshold λ_α is chosen to satisfy the false alarm rate $\alpha, \Pr_{\varphi=0,\mu}(\Lambda(Z) \geqslant \lambda_\alpha) = \alpha$. Hence, the test $\phi^*(\cdot)$ decides between $\mathscr{H}_0$ and $\mathscr{H}_1$ with the best detection probability for a bounded false alarm rate, which represents the major advantage of our approach.

The test is designed as follows, where $\|\cdot\|$ represents the Euclidean norm:

$$\phi^*(Z) = \begin{cases} \mathscr{H}_0 & \text{if} \quad \Lambda(Z) = \|P_H^{\perp}Z\|^2/\gamma_t^2 < \lambda_\alpha \\ \mathscr{H}_1 & \text{else} \end{cases} \tag{9}$$

As we will show in Section 5, this strong theoretical support also has a major impact in practice, providing results that largely outperform previous proposals. The optimal spline-based detection method developed in this section will be referred as the OSBD method in the rest of the paper.

4. Optimal sequential volume anomaly detection and isolation

In this section we introduce an optimal volume anomaly detection algorithm that has also the ability of isolating the anomaly, i.e. finding which is the particular OD flow responsible for the abnormal links traffic variation. We consider the same simplifying hypothesis as in [25], considering only "local" anomalies, namely anomalies in a single OD flow at one time. Different from Section 3, we now seek to detect and isolate an additional anomalous volume θ in one single OD flow k. This traduces into an additive change $\boldsymbol{\theta} = \theta\mathbf{r}_k$ in the SNMP measurements vector Y_t.

Instead of maximizing the probability of anomaly detection for a bounded false alarm probability, we design an algorithm that minimizes the maximum mean detection/isolation delay for an upper bounded probability of false isolation and a lower bounded mean time between consecutive false alarms, a usual measure of the false alarm rate. The mean detection/isolation delay is another crucial design criterion; indeed, the faster the detection and isolation, the faster the resolution of the problem.

The problem of detecting and isolating a volume anomaly that occurs at an unknown time t_0 is a particular case of a classical change detection/isolation problem, where the objective is to compute an alarm time T at which a change

of type $v \in \{1,2,\ldots,m\}$ in the probability distribution of a random sequence of measurements is detected. The alarm time T corresponds to the time when an anomaly in OD flow v is detected and isolated. Before going into the details of the particular algorithm, let us formally define the optimality minimax criterion that we use in the design. The optimality criterion consists of minimizing the maximum mean delay for detection/isolation, given by:

$$\overline{\mathbb{E}}(T) = \sup_{t_0 \geqslant 1, 1 \leqslant k \leqslant m} \mathbb{E}^k_{t_0}(T - t_0 | T \geqslant t_0), \tag{10}$$

where $\mathbb{E}^k_{t_0}(T - t_0 | T \geqslant t_0)$ denotes the conditional expectation of $T - t_0$ when the event $\{T \geqslant t_0\}$ is true and the kth change type occurs at time t_0, subject to the following constraints: (i) a lower bound for the mean time between two false alarms:

$$\mathbb{E}_0(T) \geqslant \upsilon \tag{11}$$

where υ is a prescribed lower bound and $\mathbb{E}_0(\cdot)$ denotes the expectation when all the measurements have the same probability density function f_0, corresponding to the anomaly-free traffic; (ii) an upper bound for the maximum probability of false isolation:

$$\max_{1 \leqslant k \leqslant m} \max_{1 \leqslant j \neq k \leqslant m} \sup_{t_0 \geqslant 1} \Pr^k_{t_0}(v = j | T \geqslant t_0) \leqslant \eta \tag{12}$$

where $\Pr^k_{t_0}(v = j | T \geqslant t_0)$ corresponds to the probability that the decision is j whereas the true change type is $k \neq j$. In brief, we require that the maximum mean detection/isolation delay given by (10) should be *as small as possible* subject to performance bounds on the mean time between consecutive false alarms and the maximum probability of false isolation.

In order to design an algorithm that verifies this minimax criterion, we shall treat the detection/isolation of a volume anomaly that occurs at an unknown time t_0 as a sequential hypothesis testing problem, where the null hypothesis $\mathscr{H}_0 = \{\text{OD flows are anomaly-free}\}(t_0 = +\infty)$ is tested against m alternatives $\mathscr{H}^k_{t_0} = \{\text{the}k\text{th OD flow presents an anomalous additional amount of traffic } \theta \text{ from time} t_0\}, k = 1,\ldots,m$. Sequential approaches are used to minimize the number of observations needed to decide among the hypotheses. The sequential hypothesis testing problem can be written as:

$$\mathscr{H}_0 : Z_t \sim \mathscr{N}(H\boldsymbol{\mu}_t, \gamma_t^2 I_r) \quad t = 1,2,\ldots \tag{13}$$

$$\mathscr{H}^k_{t_0} : \begin{cases} Z_t \sim \mathscr{N}(H\boldsymbol{\mu}_t, \gamma_t^2 I_r), & t = 1,\ldots,t_0 - 1,\ldots \\ Z_t \sim \mathscr{N}(H\boldsymbol{\mu}_t + \theta\Phi^{-\frac{1}{2}}\mathbf{r}_k, \gamma_t^2 I_r), & t = t_0,\ldots \end{cases}$$

where Z_t is the standardized measurements vector. As we did before, we can remove the nuisance parameter $\boldsymbol{\mu}$ from the detection problem. In order to only keep the anomalies-sensitive part of Z_t, we compute the residual process $U_t = WZ_t$, using a linear transformation W into a set of $r - q$ linearly independent variables. The matrix W^T is the linear rejector that eliminates the anomaly-free traffic by projection onto the left null space of H, built from the first $r - q$ eigenvectors of $P_H^\perp$ corresponding to eigenvalue 1. The rejector verifies the following relations: $WH = 0, W^TW = P_H^\perp$ and $WW^T = I_{r-q}$. Hypotheses $\mathscr{H}^k_{t_0}$ can be thus simplified by filtering the anomaly-free traffic:

$$\mathscr{H}^k_{t_0} : \begin{cases} U_t \sim \mathscr{N}(0, \gamma_t^2 I_{r-q}) & t = 1,\ldots,t_0 - 1,\ldots \\ U_t \sim \mathscr{N}(\theta\mathbf{v}_k, \gamma_t^2 I_{r-q}) & t = t_0, t_0 + 1,\ldots \end{cases}$$

where $\mathbf{v}_k = W\Phi^{-\frac{1}{2}}\mathbf{r}_k$ corresponds to the signature in the residuals of a change in OD flow k.

The recursive algorithm proposed in [42,43] perfectly fits this detection/isolation problem, with one useful feature, that of minimizing the mean number of samples needed to detect a change and decide among the different change types with bounded false alarm and false isolation rates. This algorithm is asymptotically optimal, i.e. it asymptotically minimizes the maximum mean delay for detection/isolation $\overline{\mathbb{E}}(T)$ when both the false alarm and the false isolation rates go to 0 : $\max\{\upsilon^{-1}, \eta\} \to 0$. The output of the recursive detection/isolation algorithm is twofold: (i) the alarm or stopping time T_r, which corresponds to the instant when an alarm is raised, and (ii) a decision v_r, which corresponds to the type of change that the algorithm decides for among the m possible change types:

$$\begin{aligned} &T_r = \min_{1 \leqslant k \leqslant m}\{T_r(k)\}, \quad v_r = \arg\min_{1 \leqslant k \leqslant m}\{T_r(k)\} \\ &T_r(k) = \inf\{t \geqslant 1 : s_t(k) \geqslant 0\}, \quad k = 1,\ldots,m \\ &s_t(k) = \min_{0 \leqslant j \neq k \leqslant m}[g_t(k,j) - h_{kj}], \quad k = 1,\ldots,m \end{aligned} \tag{14}$$

with $g_t(k,j) = g_t(k,0) - g_t(j,0)$. The recursive functions $g_t(k,0)$ are defined by

$$g_t(k,0) = (g_{t-1}(k,0) + u_t(k,0))_+ \tag{15}$$

$$u_t(k,0) = \log\frac{f_k(U_t)}{f_0(U_t)} \tag{16}$$

where $g_0(k,0) = 0$ for every $1 \leqslant k \leqslant m$ and $g_t(0,0) = 0$ for all t, f_0 is the probability density function of residuals under anomaly-free behavior and f_k is the probability density function of residuals $U_{t_0}, U_{t_0+1},\ldots$ after the kth type of change. The thresholds h_{kj} are chosen by the following formula:

$$h_{kj} = \begin{cases} h_d & \text{if } 1 \leqslant k \leqslant m \quad \text{and } j = 0 \\ h_i & \text{if } 1 \leqslant k,j \leqslant m \quad \text{and } j \neq k \end{cases} \tag{17}$$

where h_d and h_i are the detection and isolation thresholds. Basically, the anomaly detection is performed by comparing the m recursive functions $g_t(k,0)$ against the detection threshold h_d, while the anomaly isolation is performed by comparing the difference between these m recursive functions with the isolation threshold h_i. The stopping time $T_r(k)$ is the first time when the alternative hypothesis $\mathscr{H}^k_{t_0}$ is chosen by the sequential test as the most likely hypothesis. The stopping time T_r corresponds to the earliest of all the times $T_r(k)$ with $1 \leqslant k \leqslant m$. The detected anomaly is declared in OD flow k if the earliest of all these times was $T_r(k)$.

The choice of the detection and isolation thresholds h_d and h_i is discussed in [42], with practical comments and simulation results about the effectiveness of such thresholds. In practice, the detection threshold h_d is fixed so as to achieve the desired false alarm rate. As it follows from [42], some statistical issues of the recursive algorithm

can be solved by choosing $h_d \geqslant h_i$, and thus we will generally consider $h_i = h_d$. In other words, given the desired false alarm rate, we fix h_d and take the biggest value of h_i to minimize the false isolation rate.

A final remark about the computation of the probability density functions in (16): f_0 is nothing but a Gaussian density function of law $\mathcal{N}(0, \gamma_t^2 I_{r-q})$. In the case of f_k, the amplitude of the anomaly θ is completely unknown, and we must assume a certain distribution for it in order to correctly define f_k. Given that we are dealing with volume anomalies, it is reasonable to assume that the amplitude θ is uniformly distributed between two defined bounds θ_1 and θ_2. In this case, it is easy to see that f_k is simply a Gaussian mixture density. The bounds are introduced just for technical reasons and they can be chosen arbitrarily when dealing with volume anomalies. However, it is possible to control the sensitivity of the algorithm to detect small traffic changes instead of volume anomalies, see [43] for additional details. The choice of the bounds has little impact as regards anomaly isolation, because the signature is based on the direction of the anomaly and not on its amplitude.

The optimal sequential volume anomaly detection and isolation algorithm presented in this section will be referred as the Sequential Spline-Based (SSB) method in the rest of the paper.

5. Validation and performance evaluation

In this section we present the validation of the proposed traffic model and the evaluation of the anomaly detection/isolation algorithms using real and artificial measurements in different operational backbone networks. We first describe the datasets used in the evaluation; secondly, we validate the anomaly-free traffic model and compare its ability to infer the TM from SNMP measurements against well-known methods in the field; then we compare the performance of the OSBD method against the well-known PCA approach [25]; finally, we evaluate the SSB method in different network scenarios and compare its performance against two celebrated algorithms, one based in a sequential implementation of the PCA approach and the other based on Kalman filtering techniques [31]. In all cases we show that the performance of our algorithms in practice is in agreement with the thorough theoretical foundation.

5.1. The datasets

Data used for validation and evaluation consists of real traffic measurements from three operational networks: the Abilene network, an Internet2 backbone network at the US; the GEANT network, an European academic network; and a commercial Tier-2 ISP network. Table 1 presents the topology of each network. Abilene traffic data consists of 5′ sampled TMs collected via Netflow from the Abilene Observatory [44] and available at [45]. GEANT traffic data consists of 15′ sampled TMs, built from IGP and BGP routing information and Netflow data in [46], available at the TOTEM website [47]. The Tier-2 ISP network is a private commercial network and data is not public. Direct OD flow measurements are not available for this network. Instead, link traffic volumes are gathered every 10′ via SNMP. In order to validate our traffic model in this network topology, we compare our estimate against the well-known tomogravity estimate [5]. The tomogravity estimation method is a widely accepted method to estimate OD flow volumes from link traffic measurements, routing, and topology information.

Table 1
Network topologies and datasets.

Network	Nodes/links	ODFlows	Data	Sampling
Abilene	12–54	132	OD flows	5′
GEANT	23–74	506	OD flows	15′
Tier-2 ISP	50–168	2450	SNMP	10′

In the following evaluations, we assume that traffic flows X_t are unknown and consider the SNMP measurements Y_t as the input known data. In order to verify the stability properties of the proposed model and algorithms, two sets of measurements are used for each network topology: the "learning" dataset, used for calibration purposes, and the "testing" dataset, used to evaluate the performance of the algorithms. We shall use T_{learn} and T_{test} as the sets of time indexes associated with measurements from the learning and testing datasets respectively.

5.2. Traffic model validation and performance evaluation

The spatial SB model presented in this work is the first parametric, linear, and parsimonious model for the TM proposed in the literature. For this reason, we provide substantial evidence of its relevance and applicability in the three presented networks. Let us begin by showing that the model is stable in time and that it permits to correctly infer OD flows volume from SNMP measurements, using the Abilene and the GEANT datasets.

The learning dataset is composed of 1 h of SNMP measurements and it is used to construct the splines basis S. Given that the sampling rate in GEANT is smaller than the one used in Abilene, we interpolate intermediate measurements in the learning dataset of the former topology. The testing dataset is composed of 672 consecutive SNMP measurements. The learning dataset is measured 1 h before the testing dataset. The SB model is computed for each network using each learning dataset, following these steps: (i) the tomogravity estimate (TGE) $\hat{x}_t^{TGE}(k)$ is computed for all OD flows k and all $t \in T_{\text{learn}}$; (ii) the mean flow values $\bar{x}^{TGE}(k) = \frac{1}{\#(T_{\text{learn}})} \sum_{t \in T_{\text{learn}}} \hat{x}_t^{TGE}(k)$ are computed, where $\#(T_{\text{learn}})$ is the number of time indexes in the learning dataset; (iii) finally, the obtained mean values $\bar{x}^{TGE}(k)$ are sorted in ascending order to obtain a rough estimate of the OD flows traffic volume. In both cases the SB model is designed with cubic B-splines ($p = 3$) and 2 knots, representing small, medium-size, and large OD flows, see Fig. 2. The use of cubic splines comes directly from the shape of the curve to approximate. We use the Matlab Splines Toolbox to design q splines $\mathbf{s}(i), 1 \leqslant i \leqslant q$. The choice of cubic splines and the number of knots results in a total of

$q = (p+1)+2 = 6$ splines [34]. This clearly reflects the low-dimensionality of our anomaly-free traffic model, as q is effectively much smaller than m for both network topologies. The mean value $\bar{x}^{TGE}(k)$ of each OD flow is used to compute an estimate $\hat{\sigma}_k^2$ of σ_k^2, which leads to an estimate $\widehat{\Phi}$ of Φ, quite efficient and sufficient in practice.

The obtained calibrated model is used to infer the OD flows volume from the SNMP measurements of the testing dataset, using the SMLE estimate defined in (6). To qualify the accuracy of the SMLE estimate and to test the performance of the short learning step, we compute the relative root mean squared error (RRMSE) for every time t in the testing dataset:

$$\mathrm{RRMSE}(t) = \frac{\sqrt{\sum_{k=1}^{m} \left(x_t(k) - \hat{x}_t^{SMLE}(k)\right)^2}}{\sqrt{\sum_{k=1}^{m} x_t(k)^2}} \quad \forall t \in T_{\text{test}} \tag{18}$$

where $x_t(k)$ is the true traffic volume of OD flow k at time t and $\hat{x}_t^{SMLE}(k)$ denotes the corresponding SMLE estimate. The RRMSE has been used in previous works [6,7] as a summary of the relative estimation error for all m OD flows at every time t. Fig. 3(a) presents the temporal evolution of the RRMSE for the 672 measurements in the testing datasets for Abilene and GEANT. In both cases, the relative error remains stable in time. This result confirms the hypothesis about a certain time-invariance which has been concluded from Fig. 2.

Fig. 3(b) shows that more than 70% of the time estimation relative errors are below 10%. A deeper study of the RRMSE shows that in most cases, large RRMSE values correspond to large relative errors in the lowest-volume OD flows, which are well-known to be hard to estimate [5]. Note however that small OD flows have little impact as regards our problem of volume anomaly detection and are generally less important to estimate. The mean values of the RRMSE for the evaluation period are 8.14% for Abilene and 7.04% for GEANT. Methods proposed in the literature as "accurate" estimates present relative errors that vary between 5% and 15% [6,7], thus we conclude that the obtained results are highly satisfactory.

In the validation of the model for the Tier-2 ISP network, we compare the value of the SMLE estimate $\hat{x}_t^{SMLE}(k)$ against the tomogravity estimate $\hat{x}_t^{TGE}(k)$, using the relative root mean squared difference (RRMSD) between both estimates:

$$\mathrm{RRMSD}(t) = \frac{\sqrt{\sum_{k \in topTG-T_h} \left(\hat{x}_t^{TGE}(k) - \hat{x}_t^{SMLE}(k)\right)^2}}{\sqrt{\sum_{k \in topTG-T_h} \left(\hat{x}_t^{TGE}(k)\right)^2}} \quad \forall t \in T_{\text{test}} \tag{19}$$

Comparing all flows in (19) is not a reasonable approach. The tomogravity estimate provides quite accurate results for relatively high-volume flows, but poor for small flows [5]; we define the $topTG - T_h$ flows as those estimated flows by the tomogravity method that are stable in time and which mean value exceeds a threshold T_h. In this sense we only keep the most accurately estimated flows, removing the noisy or erratic estimates which seem to be wrongly estimated.

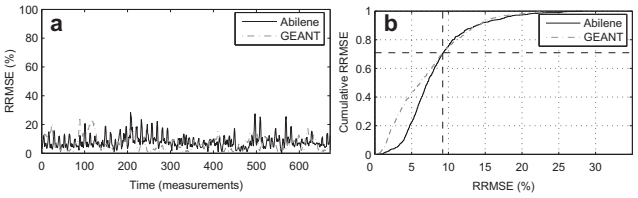


Fig. 3. (a) RRMSE(t) and (b) cumulative RRMSE (t) for 672 measurements in Abilene and GEANT.

Fig. 4 depicts the temporal evolution of the RRMSD between the TGE and SMLE estimates, for a Tier-2 ISP network. In this evaluation, we tune T_h such that 60% of the total flows are compared in the RRMSD index, which represents approximately 95% of the total traffic volume. The relative difference between TGE and SMLE is stable in time and has a mean value of 0.57%. This seems reasonable since the splines decomposition conducted in the training dataset is based on the TGE estimate. Based on our previous observations about the tomogravity estimate, we conclude that the SB model is also appropriate for this Tier-2 ISP network.

As a final validation of the SB model, we verify the Gaussian assumption for Abilene and GEANT, analyzing the residual processes U_t. Quantile-Quantile plots for two of these residual processes are plotted in Fig. 5, both for Abilene and GEANT. These residual processes clearly follow a Gaussian distribution. We also verify the Gaussian assumption by applying a Kolmogorov-Smirnov goodness-of-Fit hypothesis test to the residual processes. The acceptance rate of this test at the 5% level is 98.5% for Abilene and 97.7% for GEANT, which also confirms the Gaussian assumption.

An obvious question that arises when introducing a new TM model is how accurate this model is as regards the inference of a TM with respect to existing work in the literature. Fig. 6 presents a comparative summary of the performance of the SMLE estimate in Abilene, considering three well-known TM inference methods: a Recursive Kalman Filter Estimate (RKFE), the Simple Gravity Estimate

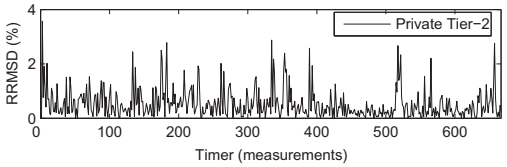


Fig. 4. RRMSD(t) for 1500 flows in a Tier-2 ISP network.

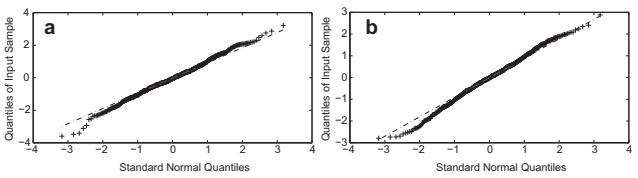


Fig. 5. QQ-plots for 2 residual processes from (a) Abilene and (b) GEANT.

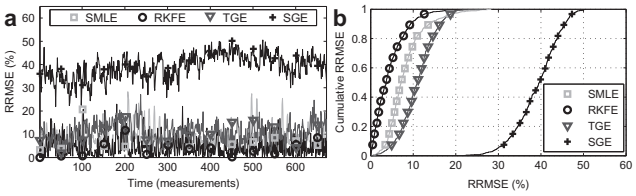


Fig. 6. (a) RRMSE (t) and (b) cumulative RRMSE (t) for 672 measurements in Abilene, for the SMLE, the RKFE, the TGE, and the SGE.

(SGE) [5], and the Tomo-Gravity Estimate (TGE) [5]. The RKFE method [8] corresponds to an enhanced extension of the recursive TM estimation method presented in [6]. This method uses a 24 h learning dataset composed of direct OD flow measurements for calibration purposes. The obtained mean values of the relative error are 8.14%, 4.48%, 11.15%, and 39.08% for the SMLE, RKFE, TGE, and SGE respectively. From Fig. 6(b) we can see that the SMLE and the RKFE produce estimation relative errors below 10% for approximately 75% and 92% of the TMs respectively, while this result drops to nearly 40% for the TGE, and to 0% for the SGE. The better performance achieved by the RKFE method has a clear explanation: the Kalman filter uses all previous SNMP measurements until time t to perform an estimated TM at time t, while the rest of the methods only use Y_t to produce an estimate $\widehat{X}_t$. The performance gain of the SMLE method w.r.t. the TGE method may not be that important, but the SB model has a clear advantage: its parametric, linear, and parsimonious structure allows to define optimal algorithms. For example, the SMLE is asymptotically optimal, i.e. it is asymptotically unbiased and efficient, which is not the case for the TGE. Presented results evidence the accuracy of the proposed spatial model w.r.t. previous highly respected work.

5.3. Numerical evaluation of the optimal detection algorithm

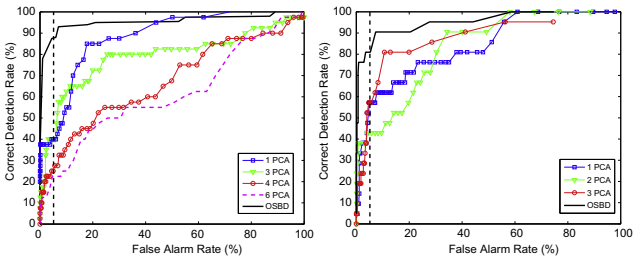
The performance of the OSBD method presented in Section 3 is compared against the performance obtained with the well-known PCA approach introduced in [25]. This method is chosen as benchmark given its relevance in the anomaly detection literature [23–25,27,29]. The PCA approach consists of a decomposition of the SNMP measurements into a principal components basis, separating traffic into a *normal subspace* that captures the anomaly-free traffic behavior, and an *anomalous subspace* that provides residuals sensitive to anomalies. This approach as presented in [25] is not designed to work on-line; instead, the analysis is performed off-line over a time window of n consecutive SNMP measurements vectors $\mathbf{Y}_{1,\dots,n} = \{Y_{t_1}, Y_{t_2}, \dots, Y_{t_n}\}^T$ (n= 1008 consecutive measurements in [25], which corresponds to 1 week of traffic). Each column in $\mathbf{Y}$ represents a time series of n samples of SNMP measurements for each network link. The normal subspace $\mathscr{S}$ corresponds to the space spanned by the first k principal components of $\mathbf{Y}_{1,\dots,n}$, namely $\mathbf{u}_{i=1,\dots,k}$, while the remaining $r-k$ components are used to build the anomalous subspace $\widehat{\mathscr{S}}$. Given $\mathscr{S}$ and $\widehat{\mathscr{S}}$, every SNMP measurements vector $Y \in \mathbf{Y}_{1,\dots,n}$ can be separated into the modeled traffic Y_{moderl} and the residual traffic Y_{residual} by simple projection onto $\mathscr{S}$ and $\widehat{\mathscr{S}}$, respectively:

$$Y = \overbrace{\mathbf{P}\mathbf{P}^T Y}^{Y_{\text{moderl}}} + \overbrace{\left(\mathbf{I} - \mathbf{P}\mathbf{P}^T\right) Y}^{Y_{\text{residual}}} \tag{20}$$

where $\mathbf{P} \in \mathbb{R}^{r\times k}$ stands for the matrix with the first k principal components $\mathbf{u}_{i=1,\dots,k}$ as column vectors and $\mathbf{P}\mathbf{P}^T$ represents the projection matrix onto the normal subspace. The anomaly detection is finally performed in the residual traffic, looking for large changes in the squared norm of residuals, $||Y_{\text{residual}}||^2$.

Let us evaluate and compare the performance of the OSBD method and the PCA approach. We shall use two testing datasets, composed of 720 consecutive SNMP measurements from the Abilene and the GEANT networks. The learning datasets for the OSBD method consist of 1 h of anomaly-free SNMP measurements, gathered 1 h before the testing datasets. In the case of the PCA approach, the method is directly applied to each complete testing dataset $\mathbf{Y}_{1,\dots,720} = \{Y_{t_1}, Y_{t_2}, \dots, Y_{t_{720}}\}^T$. For the sake of false alarm and correct detection rates evaluation, the set of "true" anomalies is manually identified in each testing dataset. Manual inspection declares an anomaly in an OD flow if the unusual deviation intensity of the guilty OD flow leads to an increase of traffic (i) larger than 1.5% of the total amount of traffic on the network and (ii) larger than 1% of the amount of traffic carried by the links routing this guilty OD flow, for each of these links. This rule is based on the conclusions about large traffic changes drawn in [33]. Hence, only large volume anomalies are considered as "true anomalies". Forty measurements of the Abilene testing dataset are affected by at least one significant volume anomaly. In the case of the GEANT testing dataset, 36 anomalous measurements are identified. Different from the PCA approach, the OSBD method is applied to the SNMP measurements of each testing dataset in an on-line fashion, sequentially running the test defined in (9) for every new "incoming" SNMP measurement $Y_{t_1}, Y_{t_2}, \dots, Y_{t_{720}}$. For the detection purpose, it is crucially important to have a good estimate of γ_t. This parameter is easily estimated from the learning dataset by using the maximum likelihood estimate of noise variance in residuals U_t [35]. Since this parameter can slowly vary in time, its value is updated during the test: at time t, if no anomaly has been declared in the last hour, γ_t is estimated by its value 1 h before.

Fig. 7 depicts the ROC curves for the OSBD and the PCA methods in the Abilene and the GEANT datasets, showing



(a) ROC Curves for Abilene (b) ROC Curves for GEANT

Fig. 7. Correct detection rate vs. false alarm rate for the OSBD method (solid line) and the PCA approach, considering a different number of k first principal components $\mathbf{u}_k$ to model the normal subspace.

the correct detection rate β for different values of the false alarm rate α, corresponding to different values of the detection threshold. In the PCA approach, a different number of k first principal components $\mathbf{u}_k$ is used to model the normal subspace. Results obtained with the PCA approach in the Abilene dataset are quite far from those obtained with the OSBD method; the PCA test presents more than 2 times lower detection rates for a reasonable false alarm rate, below 5%. For example, for a false alarm rate $\alpha = 1\%$, the OSBD method correctly detects almost 80% of the anomalies, while this value drops to nearly 40% for the best performance of the PCA approach (using 1 principal component $\mathbf{u}_1$ to model the normal subspace). Results are quite similar for the GEANT dataset, but in this case the best performance of the PCA approach is attained using 3 principal components $\mathbf{u}_{i=1,\dots,3}$ to model the normal subspace. Fig. 7 also evidences the lack of consistency of the PCA approach as regards the number of principal components used to model the anomaly-free traffic; for the same dataset, results are quite different when this number slightly varies. For the different datasets, the number of principal components that provides better results also differs, which makes it difficult to generalize results. As it is shown in recent works [29], the PCA approach has to be highly tuned for each particular dataset in order to provide reliable results, making it inapplicable in a general real scenario. In fact, the main problem with current in-house methods is the difficulty to generalize their results.

The last important observation is that the OSBD method provides highly accurate results with a remarkably short learning step, reinforcing the stability properties of the underlying parametric anomaly-free traffic model and the robustness of the approach. On the contrary, the PCA approach provides a completely data-driven model for anomaly-free traffic, resulting in the aforementioned shortcomings.

5.4. Performance evaluation of the sequential anomaly detection and isolation algorithm

Le us first demonstrate the ability of the SSB algorithm to detect and isolate an OD flow volume anomaly from SNMP measurements in two different networks, the commercial Tier-2 network and the Abilene network. Fig. 8 shows a typical realization of functions $s_t(i)$ and $g_t(i,0)$ defined in (14) and (15) respectively. Functions $s_t(i)$ are used to "monitor" the OD flows; when $s_t(i)$ exceeds the threshold 0, OD flow i is declared anomalous. The anomaly in the Tier-2 network begins at time 3660 min, and at time 1070 min in Abilene. Note that after this time, several recursive functions $g_t(i,0)$ rapidly grow in both network scenarios. Each function $g_t(i,0)$ is associated with OD flow i and when this function increases, it means that OD flow i is suspected of carrying an abnormal amount of traffic. Contrary to $g_t(i,0)$, only function $s_t(159)$ associated to anomalous OD flow 159 increases and finally exceeds the threshold 0 in the Tier-2 network. In the case of Abilene, the anomaly is correctly isolated in OD flow 87. Hence, functions $s_t(i)$ permit to isolate the anomalous OD flow among all the OD flows associated to functions $g_t(i,0)$ that have rapidly increased. The volume anomalies detected in

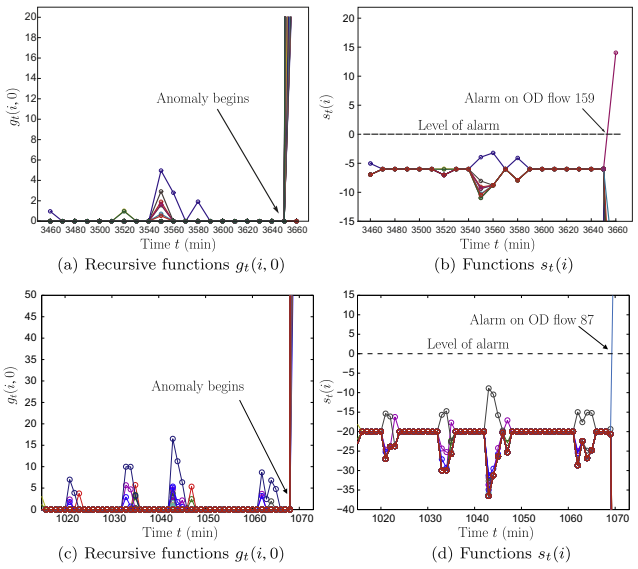


Fig. 8. Typical realizations of anomaly detection/isolation functions for a Tier-2 network (a and b) and Abilene (c and d).

the examples of Fig. 8 correspond to abrupt and massive volume augmentations, and thus functions $s_t(i)$ only need one observation to detect and isolate the anomalous OD flow. Since the underlying sampling rates of both datasets are 10′ and 5′ for the Tier-2 and the Abilene networks respectively, the detection delay corresponds to 10′ and 5′, respectively. Note however that our algorithm in not intrinsically tied to any particular sampling rate, thus this detection delay would be even shorter if the sampling rates were higher. An interesting observation of this evaluation is that the SSB algorithm achieves accurate results in both datasets, even though the respective anomaly-free traffic behaviors are quite different between these two networks.

Let us now compare the performance of the SSB algorithm to continuously detect and isolate volume anomalies in real-time against two sequential methods in the literature: the Kalman-Based method (KB) presented in [31], and a sequential implementation of the previously described PCA method that we will reference as the Sequential PCA method (SPCA). This sequential extension of the PCA approach comes from the authors of the former PCA method in [23,25], but the method was never evaluated in their anomaly detection work [25]. The idea is straightforward; the principal components and the corresponding projection matrix $\mathbf{PP}^T$ are built off-line from a certain time window $[t_1, t_n]$ of SNMP measurements $\mathbf{Y}_{1,\dots,n}$; subsequently, every new arriving measurement Y_t at time $t > t_n$ is processed on-line using this projection matrix.

The PCA and the subspace methods can also be used to detect single OD flow volume anomalies. In the subspace framework, a particular volume anomaly represents a displacement of the SNMP measurements vector Y from the normal subspace $\mathscr{S}$ into a particular direction. The authors of [25] propose to find the single largest OD flow volume anomaly that best describes this deviation by simply using a greedy search algorithm. We apply this approach to isolate volume anomalies with the SPCA method as well. The KB method is only used for volume anomaly detection as

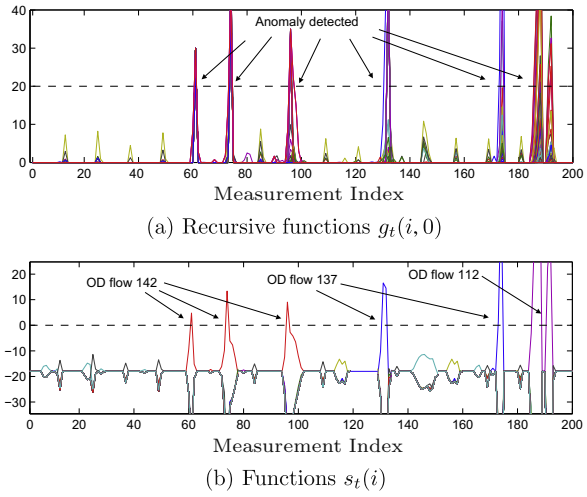


Fig. 9. On-line volume anomaly detection and isolation in Abilene, using the SSB method. The time between consecutive measurements is 5 min.

presented in [31], thus we do not intend to use it for anomaly isolation. Similar to Section 5.2, we use the enhanced extension of the recursive traffic model presented in [8] for the KB method. In order to use the SSB method to continuously detect and isolate volume anomalies, the algorithm statistics are reset to 0 after each anomaly detection, i.e., $g_t(i,0)$ is set to 0 after a change detection at time t, $\forall i = 1,\ldots,m$. Fig. 9 shows how the SSB method works on-line, continuously detecting and isolating volume anomalies in Abilene.

The testing dataset used for the evaluation consists of 864 consecutive SNMP measurements from the Abilene network. Instead of manually identifying the set of true volume anomalies, we introduce synthetic volume anomalies into this set. Indeed, in order to test the volume anomaly isolation algorithms, we need to know exactly which is the anomalous OD flow. Additionally, we need to be sure that a volume anomaly only occurs at a particular OD flow at one time, so as to fulfill the simplifying hypothesis of single OD flow anomalies considered in Section 4. We follow a similar procedure as that described in [31] to introduce 63 large synthetic volume anomalies. The basic idea of this procedure consists in extracting the long-term trend from each OD flow, adding a Gaussian noise to these "smoothed" OD flows and finally adding the synthetic volume anomalies to this "anomaly-free" smoothed dataset. These anomalies correspond to short-lived volume changes in particular single OD flows. We additionally add two short-lived volume anomalies that span multiple OD flows at the same time, in order to analyze the response of the single OD flow volume anomaly isolation algorithms in that case.

Table 2 presents the comparative performance of the three algorithms. As before, the training dataset for the SSB method consists of 1 h of anomaly-free SNMP measurements. As in [23], the training dataset for the SPCA method consists of 1 week of SNMP measurements, gathered immediately before the testing dataset and not necessarily free of volume anomalies. The PCA decomposition as proposed in [23,25] is directly applied to unfiltered data,

Table 2
Results of the detection and isolation for 864 SNMP measurements in Abilene, composed of 65 OD flow volume anomalies.

Method	Detected (%)	False alarms (%)	Isolated
SSB	93.9	1.4	90.8%
KB	90.8	1.3	n/a
SPCA ($\mathbf{u}_1$)	76.9	1.9	73.9%
SPCA ($\mathbf{u}_{i=1,\ldots,3}$)	53.9	1.7	49.2%

thus we follow this approach. Similarly to [31], the learning dataset for the KB method consists of 24 h of anomaly-free direct OD flow measurements. The detection thresholds for the three methods are set so as to achieve a false alarm rate of about 1% in the testing dataset. As we have previously stated in Section 4 and considering the observations in [42], the isolation threshold of the SSB method is set to the same value as the detection threshold, i.e., $h_i = h_d$ in Eq. (17). In order to appreciate the sensitivity of the SPCA method to the dimensionality of the normal subspace, we consider two different representations for $\mathscr{S}$, using 1 and 3 principal component(s) respectively.

The SSB method correctly detects 61 out of the 65 volume anomalies, producing a total of 12 false alarms on the 864 measurements of the testing dataset. From the 61 detected anomalies, 59 are correctly identified in the particular anomalous OD flows. The two volume anomalies that are not correctly isolated correspond to those anomalies that span multiple OD flows simultaneously. In this case the algorithm certainly produces an alarm, but the isolation step can not correctly distinguish between the anomalous OD flows. In the following section we discuss an approach to solve this problem.

Detection results are similar for the KB method, which correctly detects 59 anomalies with only 11 false alarms. Obtained results are less accurate with the SPCA approach and many anomalies go undetected. Using 1 principal component to construct the normal subspace, the SPCA method correctly detects 50 volume anomalies while triggering 16 false alarms. The detection threshold of the SPCA approach can be tuned so as to correctly detect 89% of the anomalies, but the false alarm rate climbs to approximately 6% in that case, a value almost 5 times bigger than the rest of the methods. The SPCA method has a similar problem to isolate multiple OD flows anomalies, in this case because the greedy search we used only looks for single OD flow anomalies. However, studies in [29] show that correctly identifying the anomalous OD flows with the PCA approach is inherently difficult. Results are quite poor when using 3 principal components to model $\mathscr{S}$, only detecting 35 volume anomalies and isolating 32. These results are consistent with the sensitivity analysis and the highlighted shortcomings of the PCA approach presented in [29,28].

6. Discussion

In this section we shall focus on complexity and implementation issues of the presented methods, discussing advantages and disadvantages of our proposals with re-

spect to previous works, as well as some possible extensions for the anomaly isolation algorithm.

6.1. Complexity analysis

Numerical complexity and memory storage are central issues for on-line anomaly detection. Most of previous works on network-wide anomaly detection have conceived methods for off-line detection [25,33], mining anomalies in large snapshots of data rather than treating every single measurement sequentially. These methods can be used for diagnosis of volume anomalies after their occurrence, but are rather useless for an ISP if anomaly mitigation or any other kind of countermeasure is the objective. On the contrary, our both SB methods can be used for on-line anomaly detection, and thus we should assess their complexity. Let us compare the numerical complexity of these algorithms against those used for comparison in Section 5.4, the KB method and the SPCA approach.

The OSBD method stores two matrices in memory, the matrix $\Phi^{-\frac{1}{2}}$, with $\Phi = R\Sigma R^T$, and the projection matrix $P_H^{\perp} = I_r - H(H^TH)^{-1}H^T$, with $H = \Phi^{-\frac{1}{2}}RS$. This represents a total of $3r^2/2$ variables ($P_H^{\perp}$ is symmetric), where r is the number of links in the network. The computation of $\Phi^{-\frac{1}{2}}$ and $P_H^{\perp}$ involves matrix multiplications and inversions, and thus the associated cost is $\mathcal{O}(r^3)$. There is an additional cost in the learning phase of the spline-based methods, related to the tomogravity estimate used to design the splines basis S. The cost of the tomogravity method is similar to that of the least-squares method, which implies $\mathcal{O}(m^3)$ operations to estimate an $m \times 1$ vector. All these matrices are computed off-line during the learning phase and do not affect the scalability and on-line applicability of the method. The on-line application involves three consecutive operations at every time t: the whitening of the SNMP measurements vector $Z_t = \Phi^{-\frac{1}{2}}Y_t$, the projection of the obtained vector onto the left null space of H, and the computation of the norm of this projection. All these operations have a complexity $\mathcal{O}(r^2)$.

Memory usage is similar in the case of the SSB method. The matrix $\Phi^{-\frac{1}{2}}$ is also stored, but instead of saving the projection matrix $P_H^{\perp}$ the rejector W is kept in memory, built from the first $r-q$ eigenvectors of $P_H^{\perp}$. Given the recursive structure of the SSB method, m additional variables are kept in memory, which corresponds to the m recursive functions $g_t(i,0), i = 1,\ldots,m$. For anomaly isolation purposes, the m anomaly signatures $\mathbf{v}_k \in \mathbb{R}^{(r-q)\times 1}$ are also stored. The singular value decomposition (SVD) of $P_H^{\perp}$ has a computation complexity of $\mathcal{O}(r^3)$, and as before, the construction of the splines basis involves $\mathcal{O}(m^3)$ operations. In the on-line detection/isolation phase, residuals $U_t = WZ_t$ are firstly computed and then used to update the m recursive functions $g_t(i,0)$ according to (15) and (16). Finally, the m functions $s_t(i)$ used for anomaly isolation are computed according to (14). These steps involve approximately $\mathcal{O}(r^2)$ operations for anomaly detection and $\mathcal{O}(m^2)$ additional operations for anomaly isolation.

The SPCA method keeps the symmetric projection matrix $\mathbf{C} = (I_r - \mathbf{PP}^T)$ in memory, which accounts for $r^2/2$ variables. The anomaly isolation in the SPCA method consists of a greedy search for a particular anomaly signature, each represented by a normalized column of the routing matrix $\mathbf{r}_k \in \mathbb{R}^{r\times 1}$ that must also be saved in memory. The construction of $\mathbf{PP}^T$ relies on computing the SVD of the SNMP measurements matrix $\mathbf{Y} \in \mathbb{R}^{n\times r}$, where n is the number of consecutive SNMP measurements considered, a number usually much bigger than r; for example, $n = 1008$ and $r = 49$ in [25]. This SVD has a numerical complexity of $\mathcal{O}(nr^2)$. The use of the SPCA for anomaly detection involves the projection of the SNMP measurements vector onto the anomaly subspace and the computation of the norm of this projection, with a numerical complexity of $\mathcal{O}(r^2)$. As regards anomaly isolation, the greedy search consists of constructing m possible anomaly explanations (with a cost of $\mathcal{O}(r^2)$ operations each), thus additionally adding $\mathcal{O}(mr^2)$ operations.

Finally, the KB method complexity corresponds to that of the standard Kalman filter recursive equations. We refer the reader to the original paper of the KB method [31] for additional details. The method must store in memory an $m \times m$ state transition diagonal matrix that models the evolution of the anomaly-free traffic matrix, the routing matrix R, and the noise covariance matrices associated with the observation and the evolution processes; the latter is also a diagonal matrix. This accounts for a total of $2(r^2 + m)$ variables in memory. The recursive nature of the Kalman filter implies to keep in memory two additional matrices, the $m \times r$ Kalman gain matrix and the $m \times m$ prediction error covariance matrix. The learning process of the KB method consists of a recursive Expectation Maximization (EM) approach. There are many different EM algorithms, but in all cases the resolution involves matrix operations with a numerical complexity of $\mathcal{O}(m^3)$ for the estimation of an $m \times 1$ vector. The use of the KB for on-line anomaly detection implies to update the Kalman gain, the estimation covariance error and the residual error. This involves matrix multiplications and inversions, and thus the associated cost is $\mathcal{O}(m^3)$.

Table 3 builds a raw summary of the numerical complexity and memory storage restrictions for the algorithms discussed above. Memory usage is similar in all cases, with a slightly higher requirement for the KB approach. While the SPCA method works with an $n \times r$ matrix in the learning phase, the SB and the KB methods use $m \times m$ matrices and thus they require more operations for learning issues. As regards on-line applicability, we see that the KB method is largely more expensive than the rest of the algorithms for anomaly detection, which comes directly from using the Kalman filter with large matrices. Finally, anomaly isolation involves a similar number of operations for the SSB and the SPCA methods. The important conclusion that

Table 3
Numerical complexity and memory usage for different on-line anomaly detection algorithms. On-line operations are divided into detection operations and isolation operations.

Method	No. vars. mem.	No. ops. learn	No. ops. on-line	
OSBD	$\mathcal{O}(r^2)$	$\mathcal{O}(m^3)$	$\mathcal{O}(r^2)$	n/a
SSB	$\mathcal{O}(mr)$	$\mathcal{O}(m^3)$	$\mathcal{O}(r^2)$	$\mathcal{O}(m^2)$
SPCA	$\mathcal{O}(mr)$	$\mathcal{O}(nr^2)$	$\mathcal{O}(r^2)$	$\mathcal{O}(mr^2)$
KB	$\mathcal{O}(m^2)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$	n/a

can be drawn from Table 3 is that the SB algorithms that we propose in this work have both similar or even smaller numerical complexity for on-line anomaly detection/isolation than those proposed to date.

6.2. Implementation issues

We shall now discuss some important issues related to a real implementation of the proposed algorithms in a large-scale operational network. Table 4 presents a comparative analysis of some implementation-significant features between the SB algorithms, the KB method, and the SPCA method. Let us discuss each of the compared items.

All the methods use SNMP measurements as input data for anomaly detection, making it possible, at least a-priori, to detect volume anomalies in OD flows without necessity of direct flow monitoring technology. This is a key feature regarding the development of light monitoring systems. However, the KB method needs anomaly-free (A-F) direct flow measurements for calibration purposes, loosing this advantage. The learning data for the SB methods consists of anomaly-free SNMP measurements, while the SPCA method uses SNMP measurements not necessarily free of anomalies for calibration (collected "raw" data). There is a major difference in the duration of the learning step, which we will see has important consequences. As we have shown in the evaluation section, the SB methods just need 1 h of SNMP measurements to achieve reliable results. The KB method uses 24 h of OD flow measurements to calibrate the underlying anomaly-free traffic model, and the SPCA method uses as much as 1 week of SNMP measurements to build the normal and anomalous subspaces. The use of raw SNMP measurements in the SPCA approach is certainly useful, but as it has already been shown in previous works [29,28], there is an undeniable associated risk of learning contamination, which is definitely magnified by the lengthly learning step. The remarkably short learning step of the SB methods makes it easy for network operators to calibrate the underlying spline-based model without risks of contamination, as it is quite easy to collect 1 h of SNMP measurements free of volume anomalies.

The assumptions involved in deriving the SB anomaly-free traffic model are quite strong with respect to the rest of the algorithms. Nevertheless, the validation of the SB model in three different large-scale networks shows that these assumptions are correctly verified in quite different network topologies and traffic scenarios (commercial traffic as well as research-oriented traffic). The KB method makes little assumptions on the underlying traffic model and assumes the classical Kalman filter hypotheses to be correctly verified. In practice, the Kalman filter is well-known for being robust to model imprecisions, and thus we claim that the KB assumptions are weak. The SPCA method is a pure data-driven method and makes no assumptions about traffic characteristics. However and as it is pointed out in [29], there are quite significant assumptions in the heuristics used for anomaly isolation that have no a-priori justification and can unduly trigger alarms in some OD flows much more frequently than others.

Table 4
Implementation issues in on-line anomaly detection/isolation.

Feature under comparison	SB	KB	SPCA
Input data	SNMP	SNMP	SNMP
Learning data	SNMP A-F	TM A-F	SNMP
Learning period length	1 h	24 h	1 week
Assumptions	Strong	Weak	Significant
Scalability	Yes	Poor	Yes
Dynamic routing	Partially	No	No
Non-stationary traffic	Yes	Partially	Partially
Missing data	Yes	No	Yes

The numerical complexity analysis previously performed shows that both SB methods as well as the SPCA method are easily scalable with the size of the network, while poor scalability can be expected from the KB method.

There is no discussion about the impacts of routing modifications over the SPCA method in the former papers [23,25] and a constant routing matrix is used, both in the theoretical development and in the evaluation. The authors of [31] claim that the KB method can be easily extended to work with time-varying routing matrices, but no discussion is provided on the involved challenges and current proposal does not support dynamic routing. The main challenge with routing modifications is that intradomain routing modifications can modify the incoming OD traffic flows distribution due to interdomain traffic shifts. In fact, it is well-known that hot potato routing can induce interdomain routing changes due to intradomain routing modifications. In this sense, all algorithms must be re-calibrated when an intradomain routing modification occurs, and the only methods that have a learning period length in the time-scale of a routing modification are the SB methods, thus we claim that the SB anomaly detection methods can partially support routing modifications.

A similar analysis can be done regarding the application of the methods to non-stationary OD flows. Non-stationarities in traffic flows may render the underlying anomaly-free traffic model non-longer adequate, motivating a model recalibration. The key issue is how to detect when a new recalibration must be done. In [7], authors propose a very simple heuristic to achieve this task for the underlying models of the SPCA and KB methods. The idea is to monitor the *innovation process* i_t of the traffic model, namely the difference between the measured SNMP link counts Y_t and the link counts obtained from the estimated TM, namely $\widehat{Y}_t = R\widehat{X}_t$. The decision rule is straightforward: if the innovation process is above certain threshold, a recalibration is triggered. To avoid unnecessary and expensive recalibrations due to short-lived volume anomalies, authors propose to monitor i_t during periods of 24 h, and only perform a new calibration if i_t has exceeded the threshold more than some fraction of the time. A similar heuristic could be directly applied to the SB methods. However, there are some clear drawbacks of this approach. The first problem is related to long-lived anomalies, which may not be filtered even with a 24 h window of measurements. In fact, in this case it is not possible to distinguish between an anomaly and a model that has drifted. The second problem is that the recalibration could come many hours late, seriously affecting the performance of the detection algorithm. Our SB methods have once again the

lead in this subject, due to the short and "cheap" (SNMP-based) learning period of the underlying model. A very simple heuristic to avoid drifting from an accurate model would be to proceed in a similar way to Section 5.3: simply recalibrate the model if no anomaly has been declared in the last hour. Evaluations about the temporal stability of the SB model showed that this is not necessary even for several consecutive days in the real datasets that we used. Even so, we have shown that if necessary, our method can effectively be re-calibrated every hour, and thus we claim that the SB anomaly detection methods support non-stationary traffic.

The last item we discuss concerns missing data; all algorithms use SNMP measurements as input, which has known practical limitations due to missing data and synchronization problems when collecting SNMP readings network-wide. In fact, the simultaneous collection of SNMP readings is practically impossible in very large-scale networks. The SPCA and the SB methods assume temporal independence between consecutive SNMP measurements, and thus the only impact that missing data has is a delayed verdict. In practice, it is easy to verify that all SNMP router readings are available at time t before applying the detection/isolation tests; in case there are missing readings at time t, the methods have to delay the analysis until the following time step where data is complete. As regards desynchronized readings, the problem is similar to missing data, and the best the algorithms can do is to delayed the analysis as before. Both problems condition the smallest feasible time-scale on which the proposed methods might be used, but this is an implementation issue that depends on the particular network and thus it is impossible to give an order of this smallest time-scale. A possible solution to alleviate the problem of missing and desynchronized SNMP readings is to use oversampling: oversampling is commonly used in signal processing to reduce the effect of noisy measurements. However, this analysis is beyond the scope of current paper. As regards the KB method, it strongly relies on the temporal dependence between consecutive SNMP measurements, and thus it can be heavily influenced by missing data. The Kalman filter can be modified so as to cope with missing data, but current KB implementation [31] does not support this practical limitation.

6.3. Multiple anomaly isolation

To conclude with the discussion section, we propose some possible extensions to the presented anomaly isolation algorithm. In this paper we have assumed the same simplifying hypothesis as in [25], considering only "localized" anomalies, namely anomalies in a single OD flow at a time. However, the isolation algorithm can be extended, at least in theory, to identify multiple consecutive OD flow volume anomalies. The multiple hypotheses $\mathscr{H}^k_{t_0}$ in (13) can be rewritten so as to consider multiple combinations of consecutive anomalous OD flows as additional hypotheses to test. For example, suppose that we want to detect single OD flow volume anomalies as well as volume anomalies that span two OD flows at the same time. In this case, we have to add to $\mathscr{H}^k_{t_0}$ all the hypotheses that consider a volume anomaly at OD flow i and at OD flow j at the same time, for $0 \leqslant i \neq j \leqslant m$. This accounts for $\mathscr{C}^m_2 = m!/2!(m-2)! \approx m^2/2$ additional hypotheses to test. In this case, the set of anomaly signatures is composed not only by the m single normalized columns of the routing matrix $\mathbf{r}_k, k = 1, \ldots, m$ but also by $\mathscr{C}^m_2$ matrices that include the two normalized columns of the routing matrix associated with the two anomalous OD flows. This procedure is the same as the one discussed in [25], but the idea comes from the former work of the PCA approach for fault diagnosis [21]. The problem with this approach is that the number of hypotheses to deal with, and consequently the number of decision functions $s_t(i)$ to compute grows highly and becomes very difficult to manage in a practical implementation. It is important to stress that the PCA approach [21,25] suffers from exactly the same problem as regards anomaly isolation, as the heuristics employed have a numerical complexity in the same order as our methods. The isolation of multiple consecutive anomalous OD flows is out of the scope of this paper.

7. Conclusions

In this paper we have addressed the problem of network-wide volume anomaly detection and isolation in large-scale IP networks. The following list highlights the main characteristics of the proposed solution and our major contributions to the field:

(1) Presented methods rely on coarse-grained, easily-available SNMP data to detect and isolate volume anomalies in traffic OD flows. This is a main advantage in order to develop light monitoring systems without the necessity of direct flow measurement technology, particularly in the advent of the forecast massive traffic to analyze in the near future.
(2) We have introduced an original linear, parsimonious, spline-based traffic model to describe the anomaly-free behavior of the traffic in a large-scale IP network. This spatial traffic model has several applications and advantages with respect to previous traffic matrix models: (i) being parsimonious by conception, it allows to solve the fundamentally ill-posed nature of the traffic matrix estimation problem from link SNMP measurements; (ii) it is non-data-driven and as we have verified through extensive evaluation with real data, it remains stable in time, at least for several days; (iii) the model is easy to calibrate and needs a very small amount of anomaly-free data to provide reliable results; and most importantly, (iv) this parsimonious parametric model makes it possible to remove the anomaly-free traffic from the anomaly detection problem, motivating our original approach of treating the detection and isolation of volume anomalies as a statistical change detection/isolation problem with a nuisance parameter. This a-priori simple characteristic allows to construct optimal algorithms for volume anomaly detection and isolation.
(3) We have developed different methods for volume anomaly detection and isolation with a paramount advantage with respect to previous works in the

field, that of having solid optimality properties in terms of detection mean delay, false alarm rate and false isolation rate. This represents a major breakthrough in the field and the most important contribution of the paper. We argue that optimality support is fundamental in the conception of general algorithms, not tied to any particular network or evaluation.

(4) Using extensive data from three real backbone networks we have shown that the theoretical optimality properties of the proposed algorithms are verified in practice, providing results that outperform current network-wide anomaly detection/isolation methods in a wide variety of network topologies and traffic scenarios.

(5) The complexity analysis has shown that our algorithms are more efficient than current methods to perform anomaly detection and isolation in real-time with even better results. We believe that a real implementation of our optimal algorithms could be envisaged without any modifications to current technology.

It is worth noting that the presented approaches can be easily extended to the detection and isolation of more general traffic anomalies, provided that a statistical parametric model is available. We expect that the proposed solutions in this work will stimulate in the future the development of anomaly detection algorithms with a solid theoretical background, allowing a robust growth of the network monitoring field. We believe that the results of decision theory applied to the field of network monitoring are still not sufficient and worthy to extend. This paper contributes to bridging the gap between these two fields.

Acknowledgements

We would like to thank the anonymous reviewers for their detailed comments that helped to improve the paper.

References

[1] Cisco Systems, Global IP Traffic Forecast and Methodology, 2006–2011, white paper available at <http://www.cisco.com>, 2007 – updated 2008.
[2] Cisco Systems, "The Exabyte Era", white paper available at <http://www.cisco.com>, 2007 – updated 2008.
[3] I. Cunha, F. Silveira, R. Oliveira, R. Teixeira, C. Diot, Uncovering artifacts of flow measurement tools, in: Proc. of Passive and Active Measurement Conference, 2009.
[4] P. Casas, L. Fillatre, S. Vaton, Multihour robust routing and fast load change detection for traffic engineering, in: Proc. IEEE ICC, 2008.
[5] Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, Fast accurate computation of large-scale IP traffic matrices from link load measurements, in: Proc. ACM SIGMETRICS, 2003.
[6] A. Soule, K. Salamatian, A. Nucci, N. Taft, Traffic matrix tracking using Kalman filters, in: LSNI, 2005.
[7] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, C. Diot, Traffic matrices: balancing measurements, inference, and modeling, in: Proc. ACM SIGMETRICS, 2005.
[8] P. Casas, S. Vaton, L. Fillatre, T. Chonavel, Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks, in: Proc. ITC21, 2009.
[9] C. Hood, C. Ji, Proactive network fault detection, in: Proc. INFOCOM, 1997.
[10] I. Katzela, M. Schwartz, Schemes for fault identification in communications networks, IEEE/ACM Trans. Networking 3(6) (1995) 753–764.
[11] A. Ward, P. Glynn, K. Richardson, Internet service performance failure detection, in: Performance Evaluation Review, 1998.
[12] J. Jung, B. Krishnamurthy, M. Rabinovich, Flash crowds and denial of service attacks: characterization and implications for CDNs and webs, in WWW-02, 2002.
[13] L. Xie et al., From detection to remediation: a self-organized system for addressing flash crowd problems, in: Proc. IEEE ICC-08, 2008.
[14] P. Barford, J. Kline, D. Plonka, A. Ron, A signal analysis of network traffic anomalies, in: SIGCOMM Internet Measurement Workshop, 2002.
[15] J. Brutlag, Aberrant Behavior Detection in Time Series for Network Monitoring, in: Proc. 14th Systems Administration Conference, 2000.
[16] C.M. Cheng, H. Kung, K.S. Tan, Use of spectral analysis in defense against DoS attacks, in: Proc. IEEE GLOBECOM, 2002.
[17] B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, Sketch-based change detection: methods, evaluation, and applications, in: Proc. USENIX/ACM IMC, 2003.
[18] C.C. Zou, W. Gong, D. Towsley, L. Gao, The monitoring and early detection of internet worms, IEEE/ACM Trans. on Networking 13(5) (2005) 961–974.
[19] H. Wang, D. Zhang, K. Shin, Detecting SYN flooding attacks, in: Proc. IEEE INFOCOM'02, 2002.
[20] M. Roughan, T. Griffin, M. Mao, A. Greenberg, B. Freeman, Combining routing and traffic data for detection of IP forwarding anomalies, in: ACM SIGCOMM NeTs Workshop, 2004.
[21] R. Dunia, S.J. Qin, Multi-dimensional fault diagnosis using a subspace approach, in: Proc. American Control Conference, 1997.
[22] R. Dunia, S.J. Qin, A subspace approach to multidimensional fault identification and reconstruction, Am. Inst. Chem. Eng. J. (1998) 1813–1831.
[23] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, N. Taft, Structural analysis of network traffic flows, in: Proc. ACM SIGMETICS, 2004.
[24] A. Lakhina, M. Crovella, C. Diot, Characterization of network-wide anomalies in traffic flows, in: Proc. USENIX/ACM IMC, 2004.
[25] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: Proc. ACM SIGCOMM, 2004.
[26] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: Proc. ACM SIGCOMM, 2005.
[27] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, A. Lakhina, Detection and identification of network anomalies using sketch subspaces, in: Proc. USENIX/ACM IMC, 2006.
[28] T. Ahmed, M. Coates, A. Lakhina, Multivariate online anomaly detection using kernel recursive least squares, in: Proc. IEEE Infocom, 2007.
[29] H. Ringberg, A. Soule, J. Rexford, C. Diot, Sensitivity of PCA for traffic anomaly detection, in: Proc. ACM SIGMETRICS, 2007.
[30] M. Thottan, C. Ji, Anomaly detection in IP networks, IEEE Trans. Signal Process. 51 (8) (2003) 2191–2204.
[31] A. Soule, K. Salamatian, N. Taft, Combining filtering and statistical methods for anomaly detection, in: Proc. USENIX/ACM IMC, 2005.
[32] A. Tartakovsky et al., A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods, IEEE Trans. Signal Process. 54 (9) (2006) 3372–3382.
[33] Y. Zhang, Z. Ge, A. Greenberg, M. Roughan, Network anomography, in: Proc. USENIX/ACM IMC, 2005.
[34] G. Nürnberger, Approximation by Spline Functions, Springer-Verlag, 1989.
[35] C. Rao, Linear Statistical Inference and its Applications, John Wiley & Sons, 1973.
[36] A. Wald, Tests of statistical hypotheses concerning several parameters when the number of observations is large, Trans. Am. Math. Soc. 54 (1943) 426–482.
[37] M. Fouladirad, I. Nikiforov, Optimal statistical fault detection with nuisance parameters, Automatica 41 (2005) 1157–1171.
[38] L. Fillatre, I. Nikiforov, Non-bayesian detection and detectability of anomalies from a few noisy tomographic projections, IEEE Trans. Signal Process. 55 (2) (2007) 401–413.
[39] M. Basseville, I. Nikiforov, Detection of Abrupt Changes: Theory and Applications, Prentice Hall, 1993.
[40] I. Nikiforov, A generalized change detection problem, IEEE Trans. IT 41 (1995) 171–187.
[41] T. Oskiper, H. Poor, Online activity detection in a multiuser environment using the matrix CUSUM algorithm, IEEE Trans. IT 48 (2002) 477–493.

[42] I. Nikiforov, A simple recursive algorithm for diagnosis of abrupt changes in random signals, IEEE Trans. IT 46 (7) (2000) 2740–2746.
[43] I. Nikiforov, A lower bound for the detection/isolation delay in a class of sequential tests, IEEE Trans. IT 49 (11) (2003) 3037–3046.
[44] Abilene Observatory, Available from <http://abilene.internet2.edu/observatory/>.
[45] Y. Zhang, Abilene Data, Available from <http://www.cs.utexas.edu/yzhang/>.
[46] S. Uhlig et al., Providing public intradomain traffic matrices to the research community, in: ACM Sigcomm Computer Communication Review, 2006.
[47] TOTEM Toolbox, Available from <http://totem.run.montefiore.ulg.ac.be/>.

Pedro Casas received the degree in Telecommunications Engineering from the "Universidad de la República" (UDELAR), Montevideo, Uruguay in 2005. Since November 2006 he is a Ph.D. student in Electrical Engineering at UDELAR and the Télécom Bretagne Engineering School in Brest, France. He holds a teaching and research Assistant position at UDELAR since 2001, and is a member of the Electrical Engineering Department at the Engineering School of UDELAR since 2003. His research interests are related to the statistical characterization and analysis of network traffic, network modeling, anomaly detection and performance analysis in heterogeneous networks supporting Quality of Service.

Sandrine Vaton obtained the engineering degree from Télécom Paris in 1994, a M.Sc. degree in applied probabilities from the university of Paris 6 in 1995 and a Ph.D. in signal processing from Télécom Paris in 1998. Since 1999 she is an associate professor at Télécom Bretagne in Brest, France. Her main research interests concern statistical network traffic analysis, traffic engineering, Quality of Service, and security in telecommunication networks.

Lionel Fillatre received the M.Sc. degree in decision and information engineering and the Ph.D. degree in systems optimization from the University of Technology of Troyes (UTT), France, in 2001 and 2004, respectively. From 2005 to 2007, he worked at Télécom Bretagne, Brest, France, first, as a research engineer in the Computer Science department, then as an Associate Professor in the Signal and Communications department. Since 2007, he is an Associate Professor at the Systems Modelling and Dependability Laboratory, UTT. His current research interests include statistical decision theory, signal and image processing, anomaly detection in traffic flows and information hiding in digital imagery.

Igor Nikiforov received his M.Sc. degree in automatic control from the Moscow Physical - Technical Institute in 1974, and the Ph.D. in automatic control from the Institute of Control Sciences (USSR Academy of Science), Moscow, in 1981. He joined the University of Technology of Troyes (UTT) in 1995, where he is Professor in the system modeling and dependability laboratory (LM2S), which is a part of the Institute of Charles Delaunay, FRE CNRS 2848. His scientific interests include statistical decision theory, detection/isolation of abrupt changes, fault detection/isolation/reconfiguration, signal, image processing and navigation.

Efficient Methods for Traffic Matrix Modeling and On-line Estimation in Large-Scale IP Networks

Pedro Casas*†,Sandrine Vaton*,Lionel Fillatre,‡ and Thierry Chonavel*
*TELECOM Bretagne, Brest, France - Email: name.surname@telecom-bretagne.eu
†Universidad de la República, Montevideo, Uruguay
‡Université De Technologie De Troyes, Troyes, France - Email: name.surname@utt.fr

***Abstract*—Despite a large body of literature and methods devoted to the Traffic Matrix estimation problem, the inference of traffic flows volume from aggregated data represents a key subject facing the evolution of next generation networks. This is a particular problem in large-scale carrier networks, for which efficient, accurate and stable methods for Traffic Matrix modeling and estimation are vital and challenging to conceive. In the short-term, estimation methods must be efficient and stable to allow crucial real-time tasks such as on-line traffic monitoring. In the long-term, methods must provide an accurate picture of the traffic matrix to tackle problems such as network planning, design, and dimensioning. In this paper we present and compare two efficient methods for on-line traffic matrix estimation. Based on an original parsimonious linear model for traffic flows in large-scale networks, we present a simple approach to compute an accurate traffic matrix from easily available link traffic measurements. We further extend the validation of this parsimonious model to three operational backbone networks. We analyze in depth a method to recursively estimate the traffic matrix, studying the drawbacks and omissions of the former algorithm and proposing new extensions to solve these problems. We finally perform a comparative analysis of the performance of both methods in two operational backbone networks, taking into account significant aspects such as accuracy, stability, scalability, and on-line applicability.**

***Index Terms*—Network Traffic Measurements, Modeling and Statistical Characterization, Traffic Matrix Estimation, Kalman Filtering.**

I. Introduction

Knowing and understanding the traffic that flows through a large-scale network represents a key issue in the design and engineering of the future Internet. A network-wide view of traffic flows is typically described by a traffic matrix (TM); a TM represents the volume of traffic transmitted between every pair of ingress and egress nodes of a network, also referred as the origin-destination (OD) traffic flows. The measurement of the TM is a subject of continuous debate between researchers, network operators, and technology vendors. Some of them claim that the overheads incurred in the direct measurement of the TM will become too costly and prohibitive in the future, justifying the use of aggregated data to gather its value as it has been done during the last years. This is quite a valid argument if we consider that the evolution of future access technologies and the development of optical access networks (Fiber To The Home technology) will dramatically increase the bandwidth for each end-user, triggering a brutal augmentation of the traffic to measure. Some authors forecast a value of bandwidth demand per user as high as 50 Gb/sec in 2030. On the other hand, the progress in monitoring and measurement technology of the past years make some others believe that the challenge of directly measuring the TM can be solved by improving equipment measurement capabilities and that the problem of inferring the TM from aggregated data will become obsolete. Whatever the result of this bid between increase of traffic and progress in measurement capabilities, network analysis requires efficient TM estimation methods that make use of both aggregated data and direct measurements to improve results.

Let us formally introduce the TM estimation problem. Throughout the paper, the vector $X_t = [x_t(1), .., x_t(m)]^T$ represents the value of the traffic matrix at time t, where $x_t(k)$ stands for the traffic volume of each OD flow $k = 1..m$ at time t. In a similar way, the vector $Y_t = [y_t(1), .., y_t(r)]^T$ represents the value of the links aggregated traffic volume, where $y_t(i)$ represents the total traffic volume in link $i = 1..r$ at time t. This aggregated data is available through the standard and well-known SNMP protocol, so it will be usually referred as the SNMP measurements. Given a routing matrix R, we define the traditional SNMP-TM measurement relation as:

$$Y_t = R\, X_t \tag{1}$$

where R_{ij} is equal to 1 if OD flow j traverses link i and 0 otherwise. The computation of X_t from Y_t represents a massively under constrained problem, as the number of unknown OD flows is much larger than the number of links [1]. This equation is the basis of the TM estimation problem.

A. Related Work

The problem of inferring the traffic matrix from link aggregated traffic data has been extensively studied over the past 10 years. The first approach to tackle the problem was to search for direct solutions to the ill-posed problem, introducing additional information to create additional constraints. This was achieved by TM modeling assumptions in [1], [3], deriving higher order statistics of the traffic OD flows as the additional constraints. For instance, Vardi adopted a Poisson model in [1] and Cao et al. a Gaussian model in [3]. Tebaldi et al. considered in [2] a Bayesian approach to the problem, assuming a Poisson a-priori distribution for the OD flows. A couple of years later, Medina et al. [4] showed that the basic assumptions underlying these statistical models were not

justified, and that these methods performed badly when the underlying assumptions were violated.

Additional spatial information about the TM was included into the problem, taking into account the network topology and the routing process. This encouraged the application of gravity models [5] to the estimation issue. In 2003, Zhang et al [6] made a breakthrough in the TM estimation problem, combining network tomography methods [1] with gravity models to highly improve accuracy and reduce computational complexity. This method is the well-known tomogravity estimation approach. As we will show in the obtained results, the estimation performance of the tomogravity approach can be highly improved.

A final step was achieved by considering the strong diurnal patterns found in the TM [7] into the estimation problem, together with a new strong assumption not considered before: the TM can be directly measured during short periods of time. Different works were proposed in 2004 and 2005 that exploited these assumptions [8], [9], [11]. In [8] the authors proposed a pure data-driven method to estimate the TM based on the stability of the node fanouts. [7] proposed another data-driven approach to analyze OD flows, using a Principal Component Analysis (PCA) method to capture both temporal and spatial correlations. The problem with data-driven approaches is that they are highly dependent on the data they use as input and thus results can not be generalized. The last contribution was proposed in [9], [10], where a dynamic model was adopted to capture the temporal correlation of the TM, using a Kalman Filtering approach to recursively estimate the TM. These methods make use of 24hs periods of direct OD flow measurements for calibration purposes, which can be too restrictive in a future network scenario and which limits their application to many networks that currently lack measurement technology. Even more, although they seem quite accurate and they improve previous proposals, results presented in [9], [11] showed that they can be unstable and several recalibration steps should be conducted in order to provide reliable results.

B. Contributions of the Paper

In this paper we analyze two different approaches to estimate the TM from aggregated link data. The first approach considers a spatial model for OD flows previously introduced in [13] to perform an accurate TM estimation from link data. The principal virtue of this method is that it does not require direct OD flow measurements neither to perform the estimation nor to calibrate the model. We present more evidence of the relevance and applicability of this OD flow model by extending its validation to three different operational networks: the Abilene network, the GEANT network, and a Tier-2 ISP network. The second method consists of a recursive estimation of the TM, using a Kalman filtering approach as in [9]. This method makes use of direct OD flow measurements to calibrate the subjacent flow model, using then the link data to estimate the TM value. The Kalman filtering approach is quite appealing, but the original works [9], [10] present some important drawbacks and omissions we treat in this work. By introducing new simple dynamic models we enhance the performance of the approach thus improving its applicability. Both estimation algorithms are compared in terms of relevant performance indexes namely accuracy, stability, scalability, complexity, and on-line applicability among others.

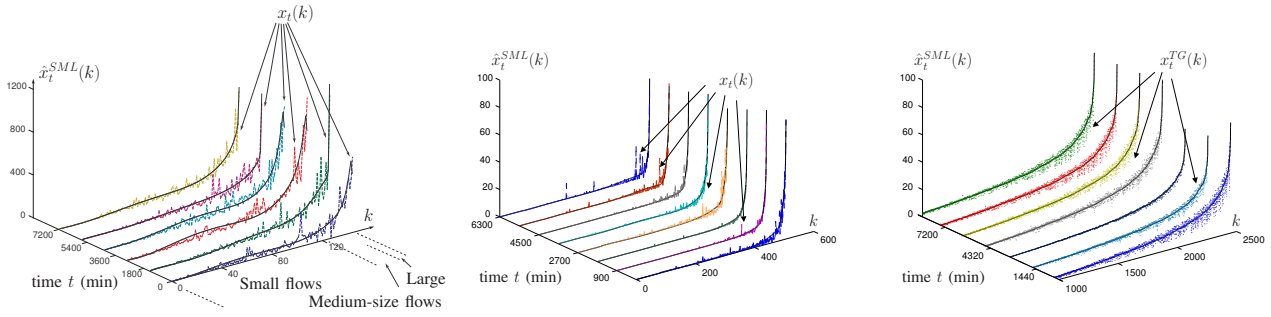
The remainder of this paper is organized as follows. In section II we recall the main aspects of the linear parsimonious OD flow model, showing how it can be applied to the traffic matrix estimation problem. Section III presents and analyses different state space OD flow models for recursive estimation of the TM, analyzing the drawbacks and omissions of previous proposals. Section IV presents the evaluation of the parsimonious model for the TM estimation problem, extending its validation to three operational backbone networks. The performance of the recursive TM estimation algorithm for different OD flow models is also evaluated, both in the Abilene and GEANT backbone networks. Finally, a comparative analysis between both estimation methods and previous proposals is presented. Section IV concludes this work.

II. Parsimonious TM Modeling and TM Estimation

In this section we recall the parsimonious linear model we have previously introduced in [13], explaining how this model can be applied to tackle the TM estimation problem. The basic idea of this model is that traffic flows X_t, sorted by OD flow volume can be decomposed at each time t over a known family of q basis functions $S = \{\mathbf{s}(1), \mathbf{s}(2), \ldots, \mathbf{s}(q)\}$, with the great virtue that $q << m$ (several orders of magnitude smaller). Therefore, we assume that X_t can be expressed as:

$$X_t = S\boldsymbol{\mu}_t + \boldsymbol{\xi}_t \tag{2}$$

where $\boldsymbol{\xi}_t$ is a white Gaussian noise with covariance matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_m^2)$ that models the natural variability of the OD TM together with the modeling errors. The vector $\boldsymbol{\mu}_t = \{\mu_t(1) \ldots \mu_t(q)\}^T$ is the unknown time varying parameter vector which describes the OD flow intensity distribution with respect to the set of vectors $\mathbf{s}(i)$. We found in [13] that the order of increasing OD flows w.r.t. their traffic volume remains stable in time for several days. Figure 1 shows the OD flows traffic for (a) the Abilene network, (b) the GEANT network, and (c) a Tier-2 ISP network, sorted in the increasing order of their volume of traffic and for different time instants t. The sorted volumes of OD flows can be interpreted as a discrete non-decreasing signal with certain smoothness. The curve obtained by interpolating this discrete signal is parameterized by using a polynomial approximation. Given the shape of this curve, a cubic splines approximation is applied. A discrete spline basis is finally built, discretizing the continuous splines according to m points uniformly chosen in the interval $[1; m]$ and rearranging them according to the OD flows sorting order. The vectors $\mathbf{s}(i)$ in S correspond to the rearranged discrete splines, which form a set of basis vectors that describe the spatial distribution of the traffic. It should be clear to the reader that this model can not be generalized to all network topologies and scenarios, but that it holds for networks with a high level of traffic aggregation (e.g., a backbone network or a large



(a) Abilene Network (b) GEANT Network (c) Tier-2 ISP Network

Fig. 1. Approximation of OD flows (dashed lines) by the spline-based model (full lines) in 3 operational networks. $\hat{x}_t^{SML}(k)$ stands for the estimated OD flow k using the spline-based model, defined in equation (6). $x_t^{TG}(k)$ is the estimated OD flow k using the tomogravity estimation method, introduced in [6].

international VPN). The dashed lines in figure 1 depict the value of each sorted OD flow $x_t(k), k = 1 \dots m$, the full lines represent the polynomial approximation of the sorted flows. In order to appreciate the time stability of this approximation, the curves are plotted for various consecutive days (at different moments of the day). Considering the SNMP-TM relation (1), the model for the link traffic is given by:

$$Y_t = G\boldsymbol{\mu}_t + \boldsymbol{\upsilon}_t, \tag{3}$$

where $G = RS$ and $\boldsymbol{\upsilon}_t \sim \mathcal{N}(0, \Phi)$, with $\Phi = R\Sigma R^T$. Since the number of columns in G is very small, the product RS and its rank can be computed very fast; therefore, we assume that G is full column rank. To simplify notation and computations, we use the whitened measurements vector Z_t:

$$Z_t = \Phi^{-\frac{1}{2}} Y_t = H\boldsymbol{\mu}_t + \gamma_t, \tag{4}$$

where $H = \Phi^{-\frac{1}{2}} G$, $\gamma_t \sim \mathcal{N}(0, I)$ and I is the identity matrix of correct dimensions. The purpose of this transformation is simply to whiten the Gaussian noise. Finally, the covariance matrix Σ is unknown. The solution consists of computing an estimate $\widehat{\Sigma}$ from a few measurements; in section IV-B we show that using just 1 hour of SNMP measurements is enough to provide proper results. Results on the estimation of $\widehat{\Sigma}$ can be found in [15]. This linear parsimonious model allows to estimate the OD flows volume X_t from easily available SNMP measurements Y_t. We propose to use a Maximum Likelihood (ML) estimation approach to compute an estimated traffic matrix. The ML estimate presents well established statistical properties [15]: it is asymptotically optimal, which means that it is asymptotically unbiased and efficient. Since the traffic linear model (4) is a Gaussian model, the ML estimate of $\boldsymbol{\mu}_t$, namely $\hat{\boldsymbol{\mu}}_t^{ML}$ corresponds to the least mean squares estimate:

$$\hat{\boldsymbol{\mu}}_t^{ML} = (H^T H)^{-1} H^T Z_t \tag{5}$$

This finally leads to the Maximum Likelihood estimate of the traffic matrix, which we will refer as the Splines-based ML (SML) estimate $\hat{X}_t^{SML}$, defined by:

$$\hat{X}_t^{SML} = S\,\hat{\boldsymbol{\mu}}_t^{ML} = \left(S(H^T H)^{-1} H^T \Phi^{-\frac{1}{2}}\right) Y_t \tag{6}$$

III. Recursive TM Estimation

The estimate $\hat{X}_t^{SML}$ presented in section II represents an estimation of X_t given the current value of SNMP measurements Y_t. In this section we present a method that not only uses Y_t to estimate X_t, but also takes advantage of the TM temporal correlation, using a set of past SNMP measurements $\{Y_{t-1}, Y_{t-2}, \dots, Y_1\}$ to compute the estimate $\hat{X}_{t|t} = \mathbb{E}(X_t | Y_t, Y_{t-1}, \dots, Y_1)$. In [9], [10], the authors use the standard Kalman filtering method [14] to recursively compute $\hat{X}_{t|t}$. We draw on the ideas of [9] as a point of departure, then analyze the weaknesses of the proposed approach, and finally extend the method to achieve more accurate and stable results.

A. A Simple State-Space Model for the Traffic Matrix

Let us consider the model that is assumed in [9], [10]. In this paper, the authors consider the traffic matrix OD flows as the hidden states of a dynamic system. A linear state space model is adopted to capture the temporal evolution of the traffic matrix, and the SNMP-TM relation (1) is used as the observation process:

$$\begin{cases} X_{t+1} &= A\,X_t + W_{t+1} \\ Y_t &= R\,X_t + V_t \end{cases} \tag{7}$$

The first equation in (7) characterizes the evolution of the OD flows X_t. A is the transition matrix that captures the dynamic behavior of the system, and W_t is an uncorrelated zero-mean Gaussian white noise that accounts both for modeling errors and randomness in the traffic flows. The second equation in (7) relates the observed links traffic Y_t to the unobserved state X_t through the routing matrix R. The measurement noise V_t is also an uncorrelated zero-mean Gaussian white noise process that models possible inconsistencies in the SNMP-TM relation. [9], [10] also assume a stationary situation where A, R, and the noise covariance matrices Q_w and Q_v are constant in time. Given this model it is possible to recursively derive the least mean squares linear estimate of X_t given $\{Y_t, Y_{t-1}, \dots, Y_1\}$, $\hat{X}_{t|t} = \mathbb{E}(X_t | Y_t, Y_{t-1}, \dots, Y_1)$ by using the standard Kalman filter (K-F) method. The Kalman filter is an efficient recursive filter that estimates the state X_t of a

linear dynamic system from a series of noisy measurements $\{Y_t, Y_{t-1}, \ldots, Y_1\}$. It consists of two distinct phases, iteratively applied: the **Prediction Phase** uses the state estimate from the previous time-step $\hat{X}_{t|t}$ to produce an estimate of the state at the current time-step $t+1$, usually known as the "predicted" state $\hat{X}_{t+1|t} = \mathbb{E}(X_{t+1}|Y_t, Y_{t-1}, \ldots, Y_1)$,

$$\begin{cases} \hat{X}_{t+1|t} &= A\,\hat{X}_{t|t} \\ P_{t+1|t} &= A\,P_{t|t}\,A^T + Q_w \end{cases} \tag{8}$$

where $P_{t|t}$ and $P_{t+1|t}$ are the covariance matrices of the estimation error $\mathrm{e}_{t|t} = X_t - \hat{X}_{t|t}$, and the prediction error $\mathrm{e}_{t+1|t} = X_{t+1} - \hat{X}_{t+1|t}$ respectively. In the **Update Phase**, measurement at the current time-step Y_{t+1} is used to refine the prediction $\hat{X}_{t+1|t}$, computing a more accurate state estimate for the current time-step $t+1$,

$$\begin{cases} \hat{X}_{t+1|t+1} &= \hat{X}_{t+1|t} + K_{t+1}\,(Y_{t+1} - R\,\hat{X}_{t+1|t}) \\ P_{t+1|t+1} &= (I - K_{t+1}\,R)\,P_{t+1|t}\,(I - K_{t+1}\,R)^T \\ & \quad + K_{t+1}\,Q_v\,K_{t+1}^T = (I - K_{t+1}\,R)\,P_{t+1|t} \end{cases} \tag{9}$$

where K_{t+1} is the optimal Kalman gain which minimizes the mean-square error $\mathbb{E}(||\mathrm{e}_{t+1|t+1}||^2)$:

$$K_{t+1} \;=\; P_{t+1|t}\,R^T(R\,P_{t+1|t}\,R^T + Q_v)^{-1} \tag{10}$$

In order to begin the Kalman filter recursion, initial conditions $\hat{X}_{0|0}$ and $P_{0|0}$ are defined. Since the value of the initial state is unknown, the initial estimate is chosen to be $\hat{X}_{0|0} = \mathbb{E}(X_0)$ and its corresponding estimation error covariance matrix $P_{0|0} = \mathbb{E}(||\mathrm{e}_{0|0}||^2)$. The calibration of matrices A, Q_w, and Q_v requires direct OD flow measurements; in [9] the authors use a 24hs period of OD flow measurements for this purpose.

In [9], the authors adopt a non-diagonal structure to the transition matrix A, while in [10] they consider a diagonal structure to A. Both choices have major impacts when using a model like (7). If we take the expected values of the right and left hand side terms in the first equation of (7) we obtain that $m_X = A\,m_X$, where $m_X = \mathbb{E}(X_t)$ denotes the average traffic matrix value. This implies that $(I - A)m_X = 0$, that is to say that m_X should be in the kernel of $I - A$. Let us consider the case where A is a diagonal matrix. In this case, the only solution to the system $(I - A)m_X = 0$ is $m_X = 0$ and obviously this condition is not satisfied by the average traffic matrix. So particularly, the first equation in (7) is false in [10], and in this context it is only valid for centered data, i.e., $m_X = 0$. Even more, our following analysis shows that using (7) without centering the data has convergence implications. On the contrary, if we consider that A is non-diagonal, it must be calibrated in such a way that $(I - A)m_X = 0$. This is essential in the model (7) as presented in [9]. In this work the authors claim that the Kalman filter must be re-calibrated every few days, when the underlying model changes, using once again direct OD flow measurements for a new 24hs period. This seems reasonable for such a particular calibration of A. As we will show in the results, this need of recalibration can be reduced with some simple corrections to the model. Let us modify the first equation in (7) in order to have a correct state space model for the case of a diagonal state transition matrix A. If we consider the variations of the OD traffic matrix X_t around its average value m_X, i.e., $X_t^{\mathrm{c}} = X_t - m_X$, the system (7) becomes:

$$\begin{cases} X_{t+1}^{\mathrm{c}} &= A\,X_t^{\mathrm{c}} + W_{t+1} \\ Y_t &= R\,X_t^{\mathrm{c}} + V_t + R\,m_X \end{cases} \tag{11}$$

The first equation in (11) is now correct for A diagonal, which corresponds to the case of modeling the centered OD flows as spatially independent AR(1) processes; even more, the equality of expected values of the left and right hand side terms holds whatever the choice of A. In this setting the model is not as sensitive to the definition of the state transition matrix A as in (7), where the only solution is to choose A non-diagonal and such that $(I - A)m_X = 0$. However, the deterministic term that appears in the observation process violates the Kalman filter assumptions; particularly, the "measurement noise" $V_t + R\,m_X$ is not a zero-mean Gaussian process. The appropriate way of treating this problem would be to center the observation process before applying the Kalman filter, using the centered observation measurements vector $Y_t^{\mathrm{c}} = Y_t - \mathbb{E}(Y_t) = Y_t - R\,m_X$. Nevertheless, we apply the Kalman filter equations to system (11) in order to appreciate the impact of using non-centered observation data when A is diagonal. Let us define $\tilde{X}_{t|t}$ as the estimate that one would obtain if the Kalman equations (8) and (9) were applied with the non-centered SNMP measurements Y_t as input. Using the Kalman filter equations, we can express both the evolution of the estimate $\hat{X}_{t|t}^{\mathrm{c}} = \mathbb{E}(X_t^{\mathrm{c}}|Y_t^{\mathrm{c}}, \ldots, Y_1^{\mathrm{c}})$ and the evolution of $\tilde{X}_{t|t}$ as:

$$\begin{array}{lll} (*) & \hat{X}_{t+1|t+1}^{\mathrm{c}} &= A\,\hat{X}_{t|t}^{\mathrm{c}} + K_{t+1}\,(Y_{t+1}^{\mathrm{c}} - RA\,\hat{X}_{t|t}^{\mathrm{c}}) \\ (**) & \tilde{X}_{t+1|t+1} &= A\,\tilde{X}_{t|t} + K_{t+1}\,(Y_{t+1} - RA\,\tilde{X}_{t|t}) \end{array} \tag{12}$$

where we have assumed the same Kalman gain in both equations as its value does not depend on the observations. If we define the error $\eta_t = \tilde{X}_{t|t} - \hat{X}_{t|t}^{\mathrm{c}}$, the difference between $(**)$ and $(*)$ can be written as:

$$\eta_{t+1} = (I - K_{t+1}\,R)A\,\eta_t + K_{t+1}\,R\,m_X \tag{13}$$

Let us assume that the Kalman filter converges; in that case, we can substitute the Kalman gain in (14) by its limit value $K = \lim_{t\to\infty} K_t$:

$$\eta_{t+1} = (I - KR)A\,\eta_t + KR\,m_X \tag{14}$$

Without loss of generality, let us suppose that $\eta_0 = 0$. We are going to prove that an error term is propagated and that the error either diverges to infinity or converges to a constant non-null value. As $\eta_0 = 0$, we can express η_t as:

$$\eta_t = \sum_{k=0}^{t-1}((I - KR)A)^k KR\,m_X, \quad \forall t > 0 \tag{15}$$

If the spectral radius of $(I - KR)A$ is greater than 1, then the error term η_t diverges to infinity. On the contrary, if the spectral radius of $(I - KR)A$ is lower than 1, then the error term η_t converges to a constant value:

$$\eta_\infty = \lim_{t\to\infty} \eta_t = (I - (I - KR)A)^{-1} KR\,m_X \tag{16}$$

This shows that, when considering a diagonal structure for the state transition matrix A in (7), not only the state space model is false but even after centering the data and explicitly introducing the mean value m_X, the Kalman filter does not converge to the real value of the traffic matrix if non-centered data Y_t is used in the filter. On the contrary, there is a gap between the real an the estimated value that is proportional to m_X (this is verified in the results in section IV-C).

B. State-Space model for centered TM variations: static mean

This problem can be easily solved in different ways. As we said, the most obvious solution would be to consider a centered observation process Y_t^c. However, we will consider a more standard approach: a deterministic term in the observation process can always be removed by adding a new deterministic state to the state model. Let us define a new state variable $U_t = [X_t^c \; m_X]^T$. In this case, (11) becomes:

$$\begin{cases} U_{t+1} &= \begin{bmatrix} A & \mathrm{O} \\ \mathrm{O} & I \end{bmatrix} U_t + \begin{bmatrix} W_{t+1} \\ \mathrm{O} \end{bmatrix} = C\,U_t + \Psi_{t+1} \\ Y_t &= \begin{bmatrix} R & R \end{bmatrix} U_t + V_t = B\,U_t + V_t \end{cases} \tag{17}$$

where O is the null matrix of accurate size. This new model has twice the number of states, augmenting the computation time and complexity of the Kalman filter. However, it presents several advantages: (i) it is not necessary to center the observations Y_t; (ii) the matrix A can be chosen as a diagonal matrix, which corresponds to the case of modeling the centered OD flows as AR(1) processes. Autoregressive models have been widely applied in the traffic matrix literature [12]; as we show in the results, obtained results with a simple AR(1) model and the K-F technique are accurate compared to the target error for standard traffic matrix estimation tools (about 10% [9], [11]) and this is clearly much easier and more stable than calibrating a non-diagonal matrix such that $(I - A)m_X = 0$; in fact, authors in [10] observe that re-calibrations are often not needed when using a diagonal transition matrix, and the results we obtain are stable during the whole evaluation period of 1 week, which is not the case in [9]; (iii) the Kalman filter estimates the mean value of the OD flows m_X, assumed constant in (17), and finally (iv) this model allows to impose a dynamic behavior to m_X, improving the estimation properties of the filter. This is exactly the step we take next.

C. Extending the model: dynamic mean

Using model (17) with the Kalman filtering technique produces quite good estimation results as we show in section IV-C. However, this model presents a major drawback: it assumes that the mean value of the OD flows m_X is constant in time. We improve (17) by adopting a simple dynamic model for m_X, in order to allow small variations of the OD flows mean value:

$$m_X(t+1) \;=\; m_X(t) + \boldsymbol{\zeta}_{t+1} \tag{18}$$

where $m_X(t)$ represents the dynamic mean value of X_t and $\boldsymbol{\zeta}_t$ is a zero-mean white Gaussian noise process with covariance matrix $Q_{\boldsymbol{\zeta}}$. This model corresponds to a random walk process, which is commonly applied to describe several dynamic models in economics, physics, etc. In this context, (17) becomes:

$$\begin{cases} U_{t+1} &= \begin{bmatrix} A & \mathrm{O} \\ \mathrm{O} & I \end{bmatrix} U_t + \begin{bmatrix} W_{t+1} \\ \boldsymbol{\zeta}_{t+1} \end{bmatrix} = C\,U_t + \Theta_{t+1} \\ Y_t &= \begin{bmatrix} R & R \end{bmatrix} U_t + V_t = B\,U_t + V_t \end{cases} \tag{19}$$

As we see in the results in section IV-C, such a simple model provides more accurate and more stable results.

IV. Evaluation and Discussion

In this section we present the evaluation of the estimation algorithms using real measurements from different operational backbone networks. We first describe the datasets used in the evaluation, then evaluate the SML estimation method and extend the validation of the splines-based model, then we evaluate the recursive Kalman filter estimation technique for the different proposed state-space models, and finally we present a comparative analysis of both algorithms.

A. The Datasets

Network	n^o nodes - links	n^o ODFlows	Data	Sampling
Abilene	12 - 54	132	OD flows traffic	5’
GEANT	23 - 74	506	OD flows traffic	15’
Tier-2 ISP	50 - 168	2450	links traffic	10’

TABLE I
Network Topologies for the Datasets.

The evaluation of the estimation algorithms is conducted using real data from two operational networks: the Abilene network, an Internet2 backbone network, and the GEANT network, a European research network. For the validation of the splines-model, we also include data from a private Tier-2 ISP network. Table I presents the topology of each network. Abilene traffic data consists of 5’ sampled TMs collected via Netflow from the Abilene Observatory [17] and available at [18]. GEANT traffic data consists of 15’ sampled TMs, built from IGP and BGP routing information and Netflow data in [19], available on the TOTEM website [20]. The Tier-2 ISP network is a private network and data is not public. Direct OD flow measurements are not available for this network. Instead, link traffic volumes are gathered each 10’ via SNMP. Using this data and a rich description of the topology, we perform a tomogravity estimation [6] of the real OD flows volume. The tomogravity method is a widely accepted method to estimate OD flow volumes from link traffic measurements and topology information with confident results. In the numerical validation of our splines model for the Tier-2 ISP network, we show that the obtained estimation results are very close to those obtained with the tomogravity estimate for this network. In the following evaluations, we assume that traffic flows X_t are just known during the calibration of the recursive Kalman algorithm and consider the SNMP measurements Y_t as the input known data. In order to verify the stability properties of the proposed models, two sets of measurements are used

for each network topology: the "learning" dataset, used for calibration purposes, and the "testing" dataset, used to evaluate the performance of the algorithms. Let T_{learn} and T_{test} be the sets of time indexes associated with measurements from the learning and testing datasets respectively.

B. Validation of the Splines Model and SML TM Estimation

In this case, both the learning and testing datasets consists of SNMP measurements. The learning dataset is composed of one hour of SNMP measurements and it is used to construct the splines basis S; the testing dataset is composed of 672 SNMP measurements. The splines-based model is computed for each network using each learning dataset, following these steps: (i) the tomogravity (TG) estimate $\hat{x}_t{}^{TG}(k)$ is computed for all OD flows k and all $t \in T_{\text{learn}}$; (ii) the mean flow values $\bar{x}\ {}^{TG}(k) = \frac{1}{\#(T_{\text{learn}})} \sum_{t \in T_{\text{learn}}} \hat{x}_t{}^{TG}(k)$ are computed, where $\#(T_{\text{learn}})$ is the number of time indexes in the learning dataset; (iii) finally, the obtained mean values $\bar{x}\ {}^{TG}(k)$ are sorted in ascending order to obtain a rough estimate of the OD flows traffic volume. The spline-based model is designed with cubic splines and 2 knots, representing small, medium-size, and large OD flows. The mean value $\bar{x}\ {}^{TG}(k)$ of each OD flow is used to compute an estimate $\hat{\sigma}_k^2$ of σ_k^2, which leads to an estimate $\hat{\Phi}$ of Φ, quite efficient and sufficient in practice.

As a global indication of the accuracy of the SML estimate and to test the performance of the short learning step, we apply the relative root mean squared error (RRMSE) for each time t in the testing dataset:

$$\text{RRMSE}(t) = \frac{\sqrt{\sum_{k=1}^{m} \left(x_t(k) - \hat{x}_t^{SML}(k)\right)^2}}{\sqrt{\sum_{k=1}^{m} x_t(k)^2}}, \quad \forall t \in T_{\text{test}} \qquad (20)$$

where $x_t(k)$ is the true traffic volume of OD flow k at time t and $\hat{x}_t^{SML}(k)$ denotes the corresponding SML estimate previously defined in (6). The RRMSE provides at each time t a summary of the relative estimation error for all m OD flows. In the validation of the model for the Tier-2 ISP network, we compare the value of the SML estimate $\hat{x}_t^{SML}(k)$ against the tomogravity estimate $\hat{x}_t^{TG}(k)$, using the relative root mean squared difference (RRMSD) between both estimates:

$$\text{RRMSD}(t) = \frac{\sqrt{\sum_{k \in \textit{topTG-}T_h} \left(\hat{x}_t^{TG}(k) - \hat{x}_t^{SML}(k)\right)^2}}{\sqrt{\sum_{k \in \textit{topTG-}T_h} \left(\hat{x}_t^{TG}(k)\right)^2}}, \quad \forall t \in T_{\text{test}} \qquad (21)$$

Comparing all flows in (20) is not a reasonable approach. The tomogravity estimate provides quite accurate results for relatively high-volume flows, but poor for small flows [6]; we define the *topTG-T_h* flows as those estimated flows by the tomogravity method that are reasonably stable in time and which mean value exceeds a threshold T_h. In this sense we only keep the most accurately estimated flows, removing the noisy or erratic estimates which seems to be wrongly estimated. Figure 2.(a) presents the temporal evolution of the RRMSE for the 672 measurements in the testing datasets for Abilene and GEANT. In both cases, the relative error remains stable in time, reinforcing the observations about time-stability of the model we drew from figure 1. Figure 2.(b)

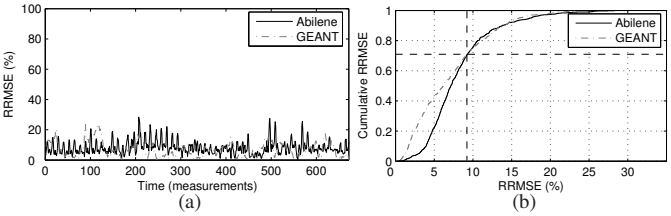


Fig. 2. (a) RRMSE(t) and (b) Cumulative RRMSE(t) for 672 measurements in Abilene and GEANT

shows that more than 70% of the time, estimation relative errors are below 10%. A deeper study of the RRMSE shows that in most cases, large RRMSE values correspond to large relative errors in the lowest-volume OD flows, which are well known to be hard to estimate [6], [11]. Note however that small OD flows have little impact on traffic engineering tasks and so are generally less important to estimate. The mean values of the RRMSE for the evaluation period are 8.14% for Abilene and 7.04% for GEANT. Methods proposed in the literature as "accurate" estimates present relative errors that vary between 5% and 15% [9], [11], so obtained results are satisfactory. Figure 3 depicts the temporal evolution

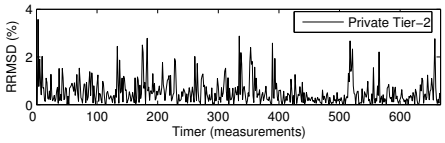


Fig. 3. RRMSD(t) for 1500 flows in a Tier-2 ISP network

of the RRMSD between the TG and SML estimates, for a Tier-2 ISP network. In this evaluation, we tune T_h such that 60% of the total flows are compared in the RRMSD index, which represents approximately 95% of the total traffic. The relative difference between the TG and the SML estimates is stable in time and has a mean value of 0.57%. Based on our previous observations about the tomogravity estimate, we conclude that the splines model is also accurate for this Tier-2 ISP network. As a final validation of the splines-model, we verify the Gaussian assumption for Abilene and GEANT. The "residuals" of measurements are analyzed, i.e., the obtained traffic after filtering the mean part $H\boldsymbol{\mu}_t$. The residuals are

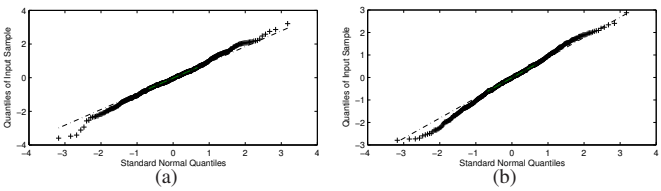


Fig. 4. QQ-plots for 2 residual processes from (a) Abilene and (b) GEANT.

obtained by projection of the whitened measurements vector Z_t onto the left null space of H. Quantile-Quantile plots for two of these residual processes are plotted in figure 4, both for Abilene and GEANT. These residual processes clearly

follow a Gaussian distribution. We also verify the Gaussian assumption by applying a Kolmogorov-Smirnov goodness-of-fit hypothesis test to the residual processes. The acceptance rate of this test at the level 5% is 98.5% for Abilene and 97.7% for GEANT, which also confirms the Gaussian assumption.

C. Recursive TM Estimation

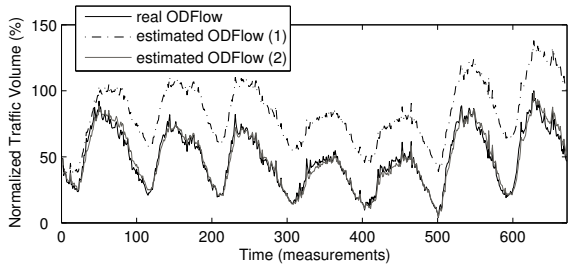


Fig. 5. Estimated OD flows using K-F for (1) model 7 and (2) model 17

The first evaluation consists of evidencing the convergence problem of the recursive TM estimation when using a model like (7) with A diagonal, as it is done in [10]. In this sense, we compare the performance of the Kalman filter using models (7) and (17). In both cases we adopt a diagonal structure for the state transition matrix A, namely an AR(1) model for each OD flow. In this evaluation and through the rest of section IV-C, the learning dataset is composed of 24hs of direct OD flow measurements X_t, as it is the case in [9]. The testing dataset consists of 1 week of SNMP measurements from the GEANT dataset, which represents 672 measurements. We also assume that the relation between X_t and Y_t is exact, that is to say $V_t = 0, \forall t$. The learning dataset is used to calibrate both models (7) and (17), namely estimate the corresponding transitions matrices and noise covariance matrices (the AR(1) parameters). We use the Yule-Walker method to compute these matrices. This method solves the Yule-Walker equations for the AR processes by means of the Levinson-Durbin recursion, see [16] for details. Figure 5 depicts the estimation of one sample OD flow with both Kalman filters; the full black curve represents the real OD flow; the dashed black curve depicts the estimated OD flow using model (7); the full gray curve depicts the estimated OD flow using model (17). In both cases, the Kalman filter properly tracks the real traffic pattern, as both curves shape are similar to the real one. However, there is a clear error-gap when using model (7), which comes from our previous analysis. Figure 6 shows the evolution of the relative

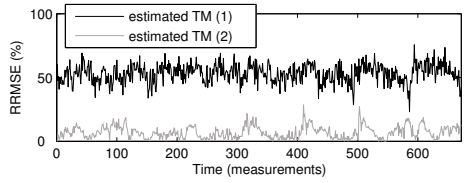
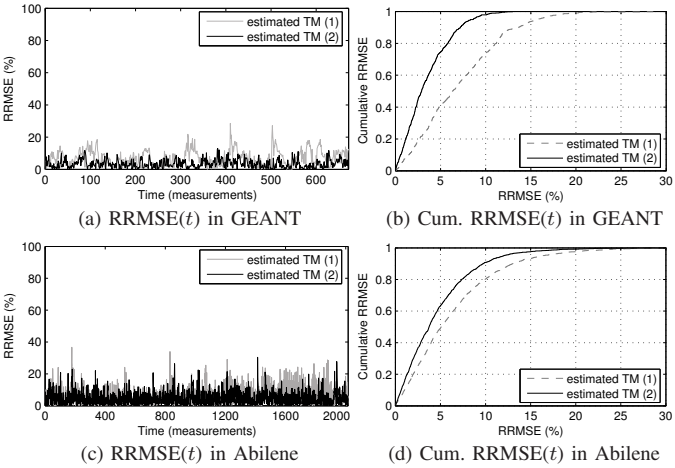


Fig. 6. RRMSE(t) for (1) model 7 and (2) model 17

estimation error RRMSD(t) for all OD flows of the TM. The mean relative error is 53.4% for model (7) and 6.2% for model (17); in both cases the error evolution is quite stable around its mean value during the whole evaluation week, giving a first evidence of the stability advantages of a diagonal transition matrix.

We now compare the estimation performance of the Kalman filter for models (17) and (19), namely assuming a constant mean value for OD flows or a random walk process, and a diagonal transition matrix in both cases. For this purpose, we consider a week of traffic in Abilene and GEANT. We consider the same assumptions adopted in the previous evaluation and calibrate the different matrices in the same way. In order to estimate the covariance matrix $Q_{\boldsymbol{\zeta}}$ of the random walk noise process $\boldsymbol{\zeta}_t$, we take the following steps: using a sliding window averaging filter we first remove the fast temporal variations from the direct OD flow measurements of the learning dataset. For each OD flow time-series, we consider the approximate derivative time-series (i.e., the difference of consecutive measurements) and compute its variance. We finally use this variance as an estimate of each diagonal element in $Q_{\boldsymbol{\zeta}}$. Figure 7 depicts the relative estimation error



(a) RRMSE(t) in GEANT (b) Cum. RRMSE(t) in GEANT

(c) RRMSE(t) in Abilene (d) Cum. RRMSE(t) in Abilene

Fig. 7. RRMSE(t) and Cumulative RRMSE(t) for 1 week of traffic in GEANT and Abilene, using (1) model 17 and (2) model 19

evolution for all TM OD flows using both models and one week of measurements in GEANT and Abilene. The cumulative RRMSE is also depicted in these figures. The obtained mean values of the relative errors are 6.20% and 4.23% in GEANT and 6.87% and 4.48% in Abilene, for models (17) and (19) respectively. We can draw two important conclusions from both evaluations: in both cases, considering a variable mean value $m_X(t)$ produces better results, both as regards accuracy and stability, as the curve of cumulative RRMSE shows a sharper growth. The second conclusion is about the advantage of correctly using a diagonal transition matrix; in all evaluations the stable evolution of the error shows that the underlying model remains valid during several days when considering such a transition matrix, a major advantage with respect to the results obtained in the former work [9]. This simple observation has a major impact on the applicability of the method in a real scenario: if the underlying model remains stable, it is not necessary to conduct periodical re-calibrations, dramatically reducing measurement overheads.

D. Comparative Analysis and Discussion

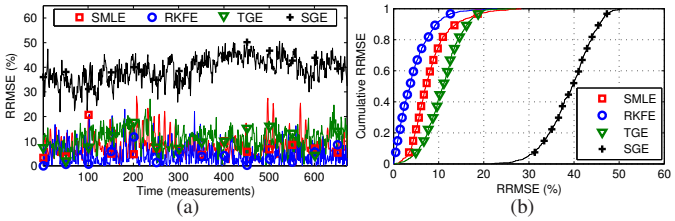


Fig. 8. (a) RRMSE(t) and (b) Cumulative RRMSE(t) for 672 measurements in Abilene, for the SMLE, the RKFE, the TGE, and the SGE.

Figure 8 presents a comparative summary of the performance of both presented methods in Abilene. The Splines-Based Maximum Likelihood Estimate (SMLE) and the Recursive Kalman Filter Estimate (RKFE) are compared against two very well known traffic matrix estimation algorithms used as baseline: the Simple Gravity Estimation method (SGE) and the Tomo-Gravity Estimation method (TGE). The obtained mean values of the relative error are 8.14%, 4.48%, 11.15%, and 39.08% for the SMLE, RKFE, TGE, and SGE respectively. From figure 8.(b) we can see that the SMLE and the RKFE produce estimation relative errors below 10% for approximately 75% and 92% of the TMs respectively, while this result drops to nearly 40% for the TGE, and to 0% for the SGE. These results allow to show the improvements of both proposed algorithms w.r.t. previous highly respected work.

To conclude with the evaluation, we present in table II a comparative analysis between the SMLE and the RKFE methods. Let us discuss each of the compared items. As regards accuracy, the RKFE presents better results, which is quite evident given its use of past data to compute the current estimate. The learning data used by the SMLE consists of pure SNMP measurements, and the method uses a remarkably short learning step. In this sense, the SMLE can be applied in networks where direct OD flow monitoring technology is not available. On the contrary, a 24hs period of direct OD flow measurements is needed to calibrate the RKFE method. As regards complexity, both algorithms are simple to implement and calibrate, specially the RKFE after the modifications introduced in this work. The considered assumptions in deriving the SMLE are quite strong compared to those adopted by the RKFE method. Nevertheless, the validation of the splines model in three different networks shows that these assumptions are correctly verified in these cases. Thanks to the underlying parsimonious model adopted in the SMLE, the method is completely scalable with the size of the network. The RKFE method does not scale with the number of OD flows in terms of computational time and memory issues, given its intrinsic recurrent characteristic and the inversion of large matrices. The scalability problem can be alleviated by implementing faster pseudo-inversion algorithms, but the problem still remains. Both algorithms can be directly applied for on-line tasks such as traffic monitoring, but the RKFE presents an interesting advantage, namely the ability to predict future values of the TM, taking advantage of the strong temporal correlation of OD flows traffic. The short learning step of the SMLE method allows its use under dynamic routing conditions, provided that the routing modifications occur at time intervals longer than 1 hour in order to allow a correct model recalibration.

Performance Index	SMLE $\hat{X}_t^{SML}$	RKFE $\hat{X}_{t\|t}$
Mean RRMSE (%) - Abilene	8.14	4.48
Mean RRMSE (%) - GEANT	7.04	4.23
Learning Data/Input Data	SNMP/SNMP	TM/SNMP
Learning Period Duration	1 hs.	24 hs.
Complexity	simple	simple
Assumptions	strong	relatively weak
Scalability	yes	partial
On-line Computation	yes	yes
Prediction Enable	no	yes
Supports Dynamic Routing	partially	no

TABLE II
COMPARATIVE PERFORMANCE OF THE SMLE AND THE RKFE.

V. CONCLUSIONS

In this paper we have revisited the TM estimation problem, dealing with important issues such as accuracy, stability, scalability, and on-line applicability among others. We have extended the validation of a previously introduced spatial model for OD flows in a large-scale network to three different operational backbone networks and showed how this model can be efficiently used in the TM estimation problem. We have introduced a simple state space model for OD flows and use it to recursively estimate the TM, using a Kalman filtering approach. A deep analysis of the drawbacks of this method as it was originally introduced allowed to better understand the stability problems of the original approach and to propose simple yet effective improvements. Both algorithms present better results than the most accepted estimation methods in the field.

REFERENCES

[1] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data", in *J. Amer. Statist. Assoc*, 91, pp. 365-377, 1996.
[2] C. Tebaldi et al, "Bayesian inference on network traffic using link count data", in *J. Amer. Statist. Assoc*, 93, pp. 557-576, 1998.
[3] J. Cao et al, "Time-varying network tomography", in *J. Amer. Statist. Assoc*, 95, pp. 1063-1075, 2000.
[4] A. Medina, K. Salamatian, S. Bhattacharyya and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *Proc. ACM SIGCOMM*, 2002.
[5] J. Kowalski and B. Warfield, "Modeling traffic demand between nodes in a telecommunications network", in *ATNAC*, 1995.
[6] Y. Zhang et al, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Load Measurements", in *Proc. ACM SIGMETRICS*, 2003.
[7] A. Lakhina et al, "Structural Analysis of Network Traffic Flows", in *Proc. ACM SIGMETRICS*, 2004.
[8] K. Papagiannaki, et al, "A Distributed Approach to Measure Traffic Matrices", in *Proc. ACM IMC*, 2004.
[9] A. Soule et al."Traffic Matrix Tracking using Kalman Filters", in *LSNI*, 2005.
[10] A. Soule, K. Salamatian and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection", in *Proc. USENIX/ACM IMC*, 2005.
[11] A. Soule et al, "Traffic Matrices: Balancing Measurements, Inference and Modeling", in *Proc. ACM SIGMETRICS*, 2005.
[12] Y. Zhang et al, "Network Anomography", in *Proc. USENIX/ACM IMC*, 2005.
[13] P. Casas, L. Fillatre and S. Vaton, "Robust and Reactive Traffic Engineering for Dynamic Traffic Demands", in *Proc. EuroNGI*, 2008.
[14] M. Hayes, "Statistical Digital Signal Processing and Modeling", J. Wiley & Sons, 1996.
[15] C. Rao, "Linear Statistical Inference and its Applications", J. Wiley & Sons, 1973.
[16] S. Kay, "Modern Spectral Estimation: Theory and Appl", Prentice-Hall, 1988.
[17] The Abilene Observatory, *http://abilene.internet2.edu/observatory/*
[18] Y. Zhang, "Abilene Dataset 04", *http://www.cs.utexas.edu/∼yzhang/*
[19] S. Uhlig et al, "Providing Public Intradomain Traffic Matrices to the Research Community", in ACM Sigcomm Computer Communication Review, 2006.
[20] TOTEM Traffic Engineering Toolbox, *http://totem.run.montefiore.ulg.ac.be/*

EQUILIBRAGE DE CHARGE EN PRESENCE D'INCERTITUDE SUR LA DEMANDE

CONTEXTE, ETAT DE L'ART ET CONTRIBUTIONS

Stratégies d'équilibrage de charge robustes et dynamiques : positionnement du problème

L'ingénierie du trafic et l'estimation de la matrice de trafic sont historiquement traités comme des sujets séparés. Cependant l'ingénierie du trafic est l'application principale de la matrice de trafic. Un certain nombre de travaux ont exploré les relations entre matrice de trafic et ingénierie du trafic [3] [24] [21] [14] [22]. Autour de ce problème j'ai coordonné en 2005 une action de recherche spécifique (SRA) du réseau d'excellence européen EuroNGI (Télécom Bretagne, HUT Helsinki, KTH Stockolm, universités de Rome 2 Tor Vergata et Rome 3 La Sapienza, France Télécom). J'ai également coordonné en 2006 un projet incitatif du GET (Groupe des Ecoles des Télécommunications, maintenant Institut Mines Télécom) DYNAM'IT dont les participants étaient Télécom Bretagne et Télécom Sud Paris.

L'approche classique est d'estimer la valeur de la matrice de trafic, puis d'optimiser l'équilibrage de charge en se basant sur la valeur estimée de la matrice. Un critère classique d'équilibrage de charge consiste par exemple à minimiser le taux maximum d'utilisation des liens. Cependant l'optimisation est faite dans un contexte d'incertitude sur la valeur de la matrice de trafic en raison d'une part des erreurs d'estimation et d'autre part des variations de la demande. De plus il est préférable d'éviter des changements du routage interne à l'AS (*Autonomous System*) trop fréquents car ils peuvent causer des instabilités et des dégradations temporaires de la QoS.

Pour configurer le routage interne on se base sur la topologie du réseau, les capacités des liens, et sur une connaissance grossière de la demande de trafic (valeurs moyennes). La responsabilité d'adapter le routage en temps réel à l'état du réseau (pannes) et à la demande de trafic effectivement observée (surcharges) retombe donc sur les administrateurs du réseau. Jusqu'à récemment on considérait que les coeurs de réseau étaient largement surdimensionnés et que les variations de trafic nécessitaient rarement de modifier le routage. On ne réagissait qu'en cas de pannes sur certains équipements ou sur certains liens. Cependant l'augmentation des débits dans l'accès avec en particulier le déploiement des technologies FTTx, les nouveaux usages de l'Internet avec la part croissante du trafic vidéo, remettent en cause ce paradigme du surdimensionnement. La seule augmentation des capacités dans le coeur en réaction à l'augmentation des débits dans l'accès n'est probablement plus pour très longtemps une solution économiquement et techniquement viable.

Deux approches antagonistes existent pour prendre en compte la variabilité dans la demande de trafic : le routage robuste et l'équilibrage de charge dynamique. Dans les approches robustes [3] [5] [21] [19] [25] l'incertitude sur la demande est prise en compte de manière proactive et intégrée directement dans le problème d'optimisation. Le résultat est donc une configuration de routage stable qui marche au mieux pour un ensemble de matrices de trafic. Cet ensemble de matrices de trafic peut être représenté comme un polyèdre d'incertitude.

Au contraire l'équilibrage de charge dynamique (MATE [14], TexCP [22]) gère la variabilité de la demande en distribuant en temps réel le trafic sur différents chemins. Pour chaque couple OD (Origine Destination) le trafic peut être routé sur différents chemins et le problème est de distribuer le trafic en temps réel sur ces chemins pour minimiser une fonction de coût.

L'avantage des approches dynamiques est qu'elles permettent d'utiliser au mieux les capacités déployées en s'adaptant en permanence à la demande de trafic. L'avantage des approches robustes est qu'elles ne nécessitent pas la mise en place de protocoles complexes, que la perte de performances en termes de taux d'utilisation des liens reste raisonnable par rapport aux approches s'adaptant au mieux à

la demande, et que les problèmes de convergence du routage ne se posent pas. En pratique les opérateurs de réseau sont réfractaires à utiliser les approches dynamiques qui ne sont donc pas ou très peu déployées dans des réseaux opérationnels et constituent essentiellement un sujet académique pour le moment.

Stratégies d'équilibrage de charge robustes et dynamiques : contributions

Nous avons étudié le problème de l'optimisation du routage dans un contexte d'incertitude sur la demande de trafic. Cette étude a été menée dans le cadre de la thèse de doctorat de Pedro Casas. Elle a donné lieu à des publications en conférences et en journaux, dont certaines ont obtenu des distinctions : [8] (NGI 2008, *Best Paper Award*) [12] (LANOMS 2009, *Best Paper Award*) [7, 10, 11, 9].

Dans nos travaux nous analysons les avantages du routage robuste par rapport aux techniques classiques qui s'adaptent à une valeur donnée de la demande sans prendre en compte l'incertitude sur celle-ci. Puis nous identifions un certain nombre de limitations du routage robuste et proposons plusieurs adaptations : le routage robuste multi-temporel (*Multi-Hour Robust Routing*, MHRR [7]), le routage robuste réactif (*Reactive Robust Routing*, RRR [9]) et le routage robuste orienté vers la QoS (*Quality of Service-based Robust Routing*, QoS-RR [11]).

L'avantage du routage robuste est qu'il fournit une configuration de routage stable qui fonctionne bien pour un ensemble de valeurs de la demande de trafic, par exemple un polyèdre d'incertitude. Disposer d'une configuration de routage unique est très intéressant car cela évite de possibles dégradations temporaires de la QoS lors des modifications du routage. Cependant un premier inconvénient de l'approche robuste est que le choix de l'ensemble d'incertitude comporte une part d'arbitraire. Un ensemble d'incertitude très grand est une garantie de conformité de la demande au modèle au prix d'une réduction de l'efficacité du routage. Au contraire une définition plus restrictive de l'ensemble d'incertitude conduit à de meilleures performances mais avec un risque plus grand que la demande réelle sorte de l'ensemble prédéfini.

Première contribution : routage robuste multi-temporel (MHRR)

Nous avons observé en étudiant des mesures de trafic sur des réseaux opérationnels (réseau international OTIP de France Télécom) qu'une source importante de variation sur la demande est facilement prédictible car elle traduit une saisonnalité de l'activité sur le réseau (variations jour/nuit, semaine/week-ends). Dans le cadre de l'action incitative du GET DYNAM'IT nous avons proposé de modéliser la demande de trafic comme un polyèdre d'incertitude qui se déforme au cours du temps et de partitionner de manère optimale cet ensemble selon l'axe du temps. De cette façon nous définissons plusieurs configurations de routage, chaque configuration robuste s'appliquant à une tranche horaire prédéfinie. L'avantage de cette approche est qu'on dispose de plusieurs configurations de routage pré-calculées et qu'une planification des heures auxquelles les changements de routage doivent être appliqués est possible. Cette approche intitulée routage robuste multi-temporel (*Multi Hour Robust Routing*, MHRR) [7] permet une définition plus restrictive du polyèdre d'incertitude et donc de meilleures performances que l'approche classique. L'optimalité du partitionnement d'un polyèdre d'incertitude pour le problème du routage robuste a été démontrée par Benameur et al. [6] suite aux travaux de DYNAM'IT. En 2005 nous avions proposé d'utiliser un modèle SARIMA (*Seasonal ARIMA*) pour modéliser et prédire la composante saisonnière du trafic [17].

Seconde contribution : routage robuste réactif (RRR)

Une autre source d'incertitude beaucoup plus difficilement prédictible est liée à l'influence du routage externe sur le trafic d'un système autonome (AS). On observe que les modifications des routes annoncées par BGP induisent un basculement du trafic d'un point d'interconnexion vers un autre et donc à un déséquilibrage de la charge interne de l'AS. Notre solution intitulée routage robuste réactif (*Reactive Robust Routing*, RRR) [9] exploite notre algorithme de détection/localisation de changements brusques dans la matrice de trafic [13]. Le principe est de détecter très rapidement les couples OD pour lesquels la demande a changé, et de proposer des modifications de routage en modifiant les proportions du trafic sur chacune des routes possibles pour ces couples OD. Nous comparons les performances de notre approche robuste et réactive à celles du routage dynamique [10] selon différents critères : d'une part la perte de

performances de la méthode robuste réactive par rapport à l'équilibrage dynamique de la charge, et d'autre part la réponse temporelle des deux approches en cas de changement brusque et important de la demande.

Troisième contribution : routage robuste orienté vers la QoS (QoS-RR)

Une autre limitation des approches robustes concerne le choix du critère d'optimisation. La complexité des techniques d'optimisation sous incertitude oblige à choisir un critère simple, comme le taux maximum d'utilisation des liens. Cependant ce critère n'est pas nécessairement un indicateur pertinent de la QoS. Minimiser le taux maximum d'utilisation des liens peut conduire, en particulier dans le cas de topologies à capacités hétérogènes, à une solution d'équilibrage de charge qui défavorise le taux moyen d'utilisation et le délai de la source à la destination. Nous introduisons un autre critère d'optimisation qui consiste à minimiser le taux moyen d'utilisation en bornant supérieurement le taux maximum d'utilisation et conduit à un problème d'optimisation sous contrainte. Puis nous améliorons notre proposition en considérant une approche multi-critère qui minimise simultanément les taux moyen et maximum d'utilisation des liens. Nous montrons par simulation que les performances obtenues la méthode de routage robuste orienté QoS (*QoS Robust Routing*, QoS-RR) en termes de délais de bout-en-bout sont meilleures que celles du routage robuste optimisant uniquement le taux maximum d'utilisation [11].

La méthode de détection/localisation des anomalies dans la demande Origine/Destination a été développée lors du séjour post-doctoral de Lionel Fillatre. Les différentes extensions du routage robuste et les comparaisons avec le routage dynamique ont été conduites par Pedro Casas dans le cadre de sa thèse de doctorat. La combinaison de ces différentes extensions conduit à une approche du routage en présence d'incertitude offrant de très bonnes performances.

Calcul de bornes sur le délai de traversée d'un domaine en présence d'incertitude sur la demande de trafic

Les travaux cités ci-dessus ont pour application l'optimisation de la QoS et en particulier du délai en intra-domaine. Alors que des solutions comme MPLS-TE ou DiffServ sont déployées depuis plusieurs années pour garantir la QoS en intra-domaine il n'existe pour le moment pas de solution satisfaisante pour la QoS inter-domaine. Pourtant beaucoup d'applications dépassent les frontières d'un seul système autonome et ont des besoins de garanties en termes de QoS (VPN, Vidéo à la Demande, TéléPrésence, Jeux sur le Réseau, ...). Par ailleurs le modèle de tarification des fournisseurs d'accès et les modèles d'interconnexion entre domaines ne permettent pas une redistribution équitable vers les opérateurs de réseaux des revenus générés par les applications ayant des besoins de QoS. Le projet intégré européen ETICS *Economics and Technologies for InterCarrier Services* [15] auquel je participe s'intéresse aux problèmes techniques et économiques posés par la QoS inter-domaine.

Dans le cadre du projet ETICS et de la thèse d'Isabel Amigo nous travaillons sur différents problèmes : la mesure de la QoS inter-domaine, la vente de services avec garanties de QoS et le partage des revenus issus de ces services, la recherche de chemins contraints inter-domaines. Pour la recherche de chemins contraints inter-domaines différents paramètres de QoS doivent être pris en compte : délai, bande passante disponible. Dans le scénario des VPN inter-domaines il existe un fort besoin de garanties en termes de délais. Ces VPNs inter-domaines utilisent généralement MPLS et posent des difficultés aux opérateurs mais ils sont plébiscités par certaines entreprises. Les banques sont par exemples demandeuses de garanties de délais pour leurs applications en finance de marchés ; en effet lors des opérations de spéculation boursière des gains considérables résultent de la capacité des opérateurs de salles de marché à prendre des positions avant la concurrence [18] [23].

Pour mettre en place des tunnels avec garantie de QoS les domaines doivent tout d'abord annoncer leurs performances. Puis un chemin de bout-en-bout satisfaisant aux contraintes de QoS est par exemple calculé avec PCE [20] et les routes sont réservées avec RSVP-TE [4] [16]. Dans ce contexte il est important que les domaines aient la capacité d'annoncer avec certitude leurs métriques. Une métrique qui nous intéresse particulièrement est le délai de traversée du domaine de routeur de bordure à routeur de bordure. Ce délai pourrait être mesuré puis annoncé ; cependant le délai peut dépendre de l'état du réseau et du trafic et n'a donc pas toujours une valeur stable. Une autre approche consiste à déterminer une borne

supérieure sur le délai de traversée de routeur de bordure à routeur de bordure et de baser les annonces sur cette borne.

Nous avons poursuivi nos études sur la prise en compte de l'incertitude dans la demande de trafic en nous intéressant au calcul de bornes sur le délai de traversée d'un AS quand la demande de trafic est de type polyèdrale [2] [1]. Nous formalisons le problème comme la maximisation d'une fonction convexe sur un polyèdre et nous présentons deux approches pour résoudre ce problème : une méthode exacte mais de complexité numérique importante, et une méthode approchée de complexité numérique beaucoup plus faible et d'erreur contrôlée. Le schéma proposé permet aux domaines d'annoncer avec certitude des bornes sur leurs métriques de délais dans un contexte d'incertitude sur la demande.

QUELQUES PUBLICATIONS INCLUES DANS LE MANUSCRIT

CASAS HERNANDEZ Pedro, FILLATRE Lionel, VATON Sandrine, NIKIFOROV Igor, Reactive Robust Routing : Anomaly Localization and Routing Reconfiguration for Dynamic Networks, Journal of Network and Systems Management (JNSM), 2011, Volume 19, Number 1, Pages 58-83

Dans cet article nous présentons un schéma complet d'adaptation du routage à l'incertitude dans la demande de trafic en prenant en compte deux sources d'inceritude : les variations régulières et les changements imprévus. Dans le scénario actuel le routage n'est pas réellement optimisé ; on compte sur le surdimensionnement des réseaux pour absorber les variations de la demande de trafic sans dégradation conséquente de la QoS. Avec l'augmentation des débits dans l'accès et dans le coeur il est probable que ce surdimensionnement des réseaux ne soit plus suffisant et qu'il soit nécessaire d'adapter le routage plus finement à la demande.

Différentes approches existent pour prendre en compte la variabilité de la demande de trafic dans la détermination des configurations de routage. Les approches robustes et stables prennent en compte de manière proactive l'incertitude sur la demande en l'intégrant directement dans le problème d'optimisation du routage. Une configuration unique de routage est calculée ; cette configuration minimise un pire cas. Au contraire les approches dynamiques tentent de s'adapter en permanence aux variations de la demande. Dans l'équilibrage de charge dynamique les proportions de trafic sur les différents chemins sont recalculées en permanence.

Dans cet article nous observons que proposer une configuration de routage unique pour l'appliquer en permanence est une approche sous-optimale. Il est préférable de disposer de plusieurs configurations de routage pré-calculées et de changer le routage, par exemple à certaines heures de la journée ; un exemple typique serait de disposer de deux configurations robustes du routage, une pour le jour et l'autre pour la nuit. Nous formalisons le problème de détermination des heures de reconfiguration du routage comme la recherche d'une partition optimale d'un polyèdre d'incertitude spatio-temporel selon l'axe des temps. Nous préconisons cette solution intitulée MHRR (*Multi Hour Robust Routing*) pour gérer les variations régulières et prévisibles de la demande.

Puis nous nous intéressons aux changements imprévus de la charge de trafic sur un AS, comme par exemple ceux qui sont liés à des modifications du routage externe. Nous présentons notre algorithme de détection/localisation de changements dans la demande. Outre ses propriétés d'optimalité (délai de détection, taux de fausse alarme/localisation), cet algorithme a la capacité de localiser les anomalies contrairement aux autres méthodes de la littérature. Cette information de localisation permet la mise en place de contre-mesures pour limiter au mieux l'impact des anomalies volumiques sur la QoS (délai de traversée de l'AS). Une fois l'anomalie détectée et localisée le polyèdre d'incertitude est redéfini en prenant en compte la direction spatiale de l'incertitude et une nouvelle configuration robuste de routage est appliquée. Une fois l'anomalie terminée un retour au schéma de routage initial est possible.

Le schéma complet intitulé RRR (*Reactive Robust Routing*) est une combinaison de méthodes proactives et réactives pour gérer les variations prévisibles et imprévisibles de la demande. Ce schéma offre de bien meilleures performances que les approches robustes classiques qui produisent une configuration de routage unique. Une comparaison avec les schémas d'équilibrage dynamique de la charge comme TeXCP est également réalisée. En particulier le schéma RRR ne souffre pas des problèmes de convergence lente bien connus des approches dynamiques puisqu'il réagit immédiatement en cas de détection d'un change-

ment brusque dans la demande de trafic.

CASAS HERNANDEZ Pedro, LARROCA Federico, ROUGIER Jean-Louis, VATON Sandrine, Taming traffic dynamics : analysis and improvements, Computer Communications (COMCOM), volume 35, issue 5, pages 565-578, march 2012

L'objectif de cet article qui est issu d'une collaboration entre Télécom Bretagne, Télécom ParisTech et IIE/UdelaR (Uruguay) est de comparer deux approches en routage : l'équilibrage de charge dynamique (DLB, *Dynamic Load Balancing*) et l'équilibrage de charge robuste réactif (RRLB, *Reactive Robust Load Balancing*). Nous nous intéressons à deux critères en particulier : la perte de performances de l'équilibrage de charge robuste par rapport à l'équilibrage de charge dynamique, et la convergence de l'équilibrage de charge dynamique en cas de changements importants dans la demande.

Nous montrons qu'une approche robuste non réactive est clairement sous-optimale et qu'une forme de réactivité est nécessaire, qu'il s'agisse de RRLB ou de DLB. Puis nous proposons différentes améliorations des approches RRLB et DLB. Pour l'approche RRLB nous proposons de choisir comme critère d'optimisation le taux moyen d'utilisation (en bornant le taux maximum d'utilisation) plutôt que le critère habituel du taux maximum d'utilisation des liens. Nous montrons que ce nouveau critère permet d'obtenir de bien meilleurs délais de bout-en-bout que le critère habituel. On reproche souvent à l'approche DLB ses problèmes de convergence lente en cas de changement brusque et important dans le trafic. Nous montrons qu'il est cependant possible de pallier à ce défaut et introduisons une approche *restart* qui permet à l'algorithme de s'ajuster beaucoup plus vite aux nouvelles conditions de trafic en cas de changement brusque.

AMIGO Isabel, VATON Sandrine, CHONAVEL Thierry, LARROCA Federico, Maximum Delay Computation for Interdomain Path Selection, International Journal of Network Management (IJNM), volume 22, issue 2, pages 162-179, march/april 2012

Dans cet article nous traitons le problème de calcul d'une borne sur le délai de traversée d'un AS quand la demande de trafic sur cet AS est de type polyèdrale. Le problème est formalisé puis deux méthodes de résolution sont proposées : une méthode exacte de complexité numérique élevée, et une méthode approchée de complexité numérique plus faible.

La demande de trafic est supposée être polyèdrale. Il peut s'agir par exemple d'un modèle de Hose, populaire pour caractériser la demande de trafic d'un VPN. Par souci de simplification le délai d'attente sur chacun des liens est modélisé selon une file M/M/1 mais tout autre fonction approximable par une fonction linéaire par morceaux pourrait être utilisée. La formalisation du problème de calcul de borne sur le délai d'un chemin conduit à la *maximisation* d'une fonction *convexe* sur un polyèdre.

Maximiser une fonction convexe sur un polyèdre est un problème difficile, d'autant plus que la fonction à maximiser n'est pas strictement convexe. Nous reparamétrons le problème de façon à se ramener à une fonction objectif strictement convexe. Cette fonction objectif doit être optimisée sur un polyèdre obtenu par projection du polyèdre initial. La solution du problème d'optimisation est un point extrêmal du polyèdre projeté.

Une méthode de résolution exacte est proposée ; cette méthode se base sur l'énumération des points extrêmaux du polytope. La solution exacte a une complexité numérique élevée car le problème de détermination des points extrêmaux d'un polyèdre n'a pas de solution connue en temps polynomial. Cette méthode est validée numériquement sur une instance de petite taille du problème (réseau Abilène). Puis une méthode approchée est proposée ; cette méthode se base sur une approximation linéaire par morceaux de la fonction à maximiser. La méthode approchée conduit à une solution arbitrairement proche de la solution exacte avec une complexité numérique bien moindre.

PERSPECTIVES

De nouveaux travaux ont été initiés lors des séjours sabbatiques du Dr Hervé Kérivin (Clemson University) et de sa doctorante Erin Doolittle. Ces travaux s'intéressent à l'optimisation multi-critère du routage en présence d'incertitude sur la demande de trafic. Plus précisément le travail concerne la détermination des points Pareto-optimaux dans le cas où la demande de trafic est de type polyèdrale. Le cas de deux fonctions objectif a été résolu, la résolution du problème dans le cas de trois fonctions objectif est en cours.

L'idée de cette approche est de ne pas pondérer les fonctions objectif pour se ramener à un critère unique. Au contraire l'ensemble des points Pareto-optimaux est déterminé et on reporte la décision sur les critères à privilégier après la détermination des points optimaux plutôt que de la prendre avant. Le cas où le domaine d'incertitude est ellipsoïdal plutôt que polyèdral est également étudié ; cela conduit à un problème d'optimisation moins linéaire mais toujours multi-objectifs.

Ce travail est réalisé dans le cadre de la thèse de doctorat d'Erin Doolittle suite à la collaboration avec Télécom Bretagne. Par la suite nous allons analyser par des simulations intensives sur différents réseaux comment les solutions efficaces peuvent être interprétées en termes de QoS, d'équilibrage de la charge, et de prise de décisions sur le réseau.

Références

[1] I. Amigo, S. Vaton, T. Chonavel, and F. Larroca. Maximum delay computation under traffic matrix uncertainty and its application to interdomain path selection. In *INOC 2011 : International Network Optimization Conference*, Heidelberg, Germany, 2011.

[2] I. Amigo, S. Vaton, T. Chonavel, and F. Larroca. Maximum delay computation for interdomain path selection. *International Journal of Network Management*, 22(2) :162–179, march/april 2012.

[3] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands : understanding fundamental tradeoffs. In *SIGCOMM*, 2003.

[4] Awduche, D. and Berger, L. and Li, T. and Srinivasan, V. and Swallow, G. RSVP-TE : Extensions to RSVP for LSP Tunnels (IETF RFC 3209). `http://www.ietf.org/rfc/rfc3209.txt`.

[5] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. *Optimization and Engineering*, 6 :283–313, 2005.

[6] W. Ben-Ameur and M. Zotkiewicz. Robust routing and optimal partitioning of a traffic demand polytope. *International Transactions in Operational Research*, 18 :307–333, 2011.

[7] P. Casas, L. Fillatre, and S. Vaton. Multi hour robust routing and fast load change detection for traffic engineering. In *ICC*, 2008.

[8] P. Casas, L. Fillatre, and S. Vaton. Robust and reactive traffic engineering for dynamic traffic demands. In *NGI 2008 : 4th EuroNGI Conference on Next Generation Internet Networks*, 2008.

[9] P. Casas, L. Fillatre, S. Vaton, and I. Nikiforov. Reactive robust routing : Anomaly localization and routing reconfiguration for dynamic networks. *Journal of Network and System Management*, 19(1) :58–83, 2011.

[10] P. Casas, F. Larroca, J.L. Rougier, and S. Vaton. Robust routing versus dynamic load balancing : a comprehensive study and new directions. In *DRCN'09 : 7th International Workshop on the Design of Reliable Communication Network*, 2009.

[11] P. Casas, F. Larroca, J.L. Rougier, and S. Vaton. Taming traffic dynamics : Analysis and improvements. *Computer Communications*, 35 :565–578, march 2012.

[12] P. Casas, F. Larroca, and S. Vaton. Robust routing mechanisms for intradomain traffic engineering in dynamic networks. In *LANOMS'09 : 6th Latin-American Network Operations and Management Symposium*, 2009.

[13] P. Casas, S. Vaton, L. Fillatre, and I. Nikiforov. Optimal volume anomaly detection and isolation in large-scale ip networks using coarse-grained measurements. *Computer Networks*, 54(11) :1750–1766, 2010.

[14] A. Elwalid, C. Jin, S. Low, and I. Widjaja. Mate : Mpls adaptive traffic engineering. pages 1300–1309, 2001.

[15] European Commission, Information Society Technologies, Seventh Framework Program. ETICS : Economics and Technologies for InterCarrier Services. https://www.ict-etics.eu/.

[16] Farell, A. and Ayyangar, A. and Vasseur, J.P. nter-Domain MPLS and GMPLS Traffic Engineering – Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions (IETF RFC 5151). http://www.ietf.org/rfc/rfc5151.txt.

[17] L. Fillatre, D. Marakov, and S. Vaton. Forecasting seasonal traffic flows. In *EuroNGI Workshop on QoS and Traffic Control*, 2005.

[18] Giacalone, S. and Ward, D. and Drake, J., and Atlas, A. and Previdi, S. OSPF Traffic Engineering (TE) Express Path (IETF Draft). http://tools.ietf.org/html/draft-giacalone-ospf-te-express-path-01.html.

[19] A. Gunnar and M. Johansson. Robust load balancing under traffic uncertainty - tractable models and efficient algorithms. *Telecommunication Systems*, 48(1-2) :93–107, 2011.

[20] IETF. PCE : Path Computation Element. http://datatracker.ietf.org/wg/pce/.

[21] M. Johansson and A. Gunnar. Data-driven traffic engineering : techniques, experiences and challenges. In *Broadband Communications, Networks and Systems, BROADNETS*, pages 1–10, 2006.

[22] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny. Walking the tightrope : Responsive yet stable traffic engineering. In *In Proc. ACM SIGCOMM*, 2005.

[23] Popsi : the future now. Fiber-optic transatlantic cable could save milliseconds, millions by speeding data to stock traders. http://www.popsci.com/technology/article/2011-04/new-transatlantic-cable-will-speed-information-exchange-price.

[24] M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, 2003.

[25] H. Wang, H. Xie, L. Qiu, Y.R. Yang, Y. Zhang, and A. Greenberg. Cope : traffic engineering in dynamic networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, 2006.

J Netw Syst Manage (2011) 19:58–83
DOI 10.1007/s10922-010-9182-9

Reactive Robust Routing: Anomaly Localization and Routing Reconfiguration for Dynamic Networks

Pedro Casas · Lionel Fillatre · Sandrine Vaton · Igor Nikiforov

Received: 30 November 2009 / Accepted: 6 October 2010 / Published online: 4 November 2010

Abstract This paper presents a novel approach to deal with dynamic and highly uncertain traffic in dynamic network scenarios. The Reactive Robust Routing (RRR) approach is introduced, a combination of proactive and reactive techniques to improve network efficiency and robustness, simplifying network operation. RRR optimizes routing for normal-operation traffic, using a time-varying extension of the already established Robust Routing technique that outperforms the stable approach. To deal with anomalous and unexpected traffic variations, RRR uses a fast anomaly detection and localization algorithm that rapidly detects and localizes abrupt changes in traffic flows, permitting an accurate routing adaptation. This algorithm presents well-established optimality properties in terms of detection/localization rates and localization delay, which allows for generalization of results, independently of particular evaluations. The algorithm is based on a novel parsimonious model for traffic demands which allows for detection of anomalies using easily available aggregated-traffic measurements, reducing the overheads of data collection.

P. Casas (✉)
Universidad de la República, J. Herrera y Reissig 565, CP 11300 Montevideo, Uruguay
e-mail: pcasas@fing.edu.uy

L. Fillatre · I. Nikiforov
ICD/LM2S, Université de Technologie de Troyes, 12 rue Marie Curie, BP 2060,
10010 Troyes Cedex, France
e-mail: lionel.fillatre@utt.fr

I. Nikiforov
e-mail: igor.nikiforov@utt.fr

S. Vaton
Télécom Bretagne, Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France
e-mail: sandrine.vaton@telecom-bretagne.eu

Keywords Traffic uncertainty · Robust routing and load balancing · Traffic modeling · Network monitoring · Optimal anomaly detection and localization

1 Introduction

The performance of the Internet itself depends, in large measure, on the operation of the underlying routing protocols. Today's routing protocols in IP networks compute routing configurations based on network topology and some rough knowledge of traffic demands (e.g., worst-case traffic, average traffic, long-term forecasts), without regard to current traffic load on routers or even possible traffic misbehaviors that often arise. Over-provisioning has been so far the main reason of success for such a "naive" approach. However, routing optimization is becoming increasingly difficult due to the dynamic nature and the increasing uncertainty of current traffic demands. The overall IP traffic is expected to grow from 6.6 exabytes per month in 2007 to nearly 29 exabytes per month by 2011, more than quadrupling in less than a half decade [1]. Simultaneously, the evolution of access technologies and the development of optical access networks (e.g. Fiber To The Home technology) is dramatically increasing the bandwidth for end-users, imposing serious and unforeseen problems at the core network, so far assumed infinitely provisioned. In this near future scenario, traditional over-provisioning may no longer be an economically viable solution to handle dynamic traffic demands.

In this paper we address the problem of routing optimization, automatic routing reconfiguration, and load balancing in a single Autonomous System (AS). The traffic demand in an AS network is typically described by a traffic matrix (TM), which captures the amount of traffic transmitted between every pair of ingress and egress nodes of the network, so called the Origin-Destination (OD) traffic flows. OD flows present two different behaviors: on the one hand, a stable and predictable behavior due to normal usage patterns (e.g. daily traffic demand fluctuation); on the other hand, a highly dynamic and unpredictable behavior due to unexpected events, such as network equipment failures, flash crowd occurrences, security threats (e.g. denial of service attacks), external routing changes (e.g. inter-AS routing through BGP), and new spontaneous overlay services (e.g. P2P applications for real-time content delivery). We shall use the term *volume anomaly* [2] to describe these unexpected events, characterized by large and sudden link load changes.

Recent works [3–5] have proposed a novel solution to the routing optimization under traffic uncertainty problem: the Stable Robust Routing (SRR) approach. In a robust perspective of routing optimization, demand uncertainty is taken into account directly within the optimization problem, computing a single routing configuration for all demands within some *uncertainty set*. While this routing configuration is not optimal for any single TM within the set, it minimizes the worst case performance over the whole set. In this sense, SRR provides worst-case performance bounds for all possible traffic variations within the uncertainty set. The obtained robust routing configuration is usually applied during long periods of time (e.g. daily routing), avoiding the potential instabilities induced by routing changes. The SRR approach can be therefore seen as a *proactive* technique to deal with dynamic traffic. It can

handle variable traffic demands up to a certain limit, given by the size of the uncertainty set. However, using a SRR algorithm to address both normal-operation traffic as well as volume anomalies is an inefficient strategy: it is clear that a single routing configuration cannot be adequate for both situations. On the contrary, a *reactive* approach could be used as a complementary strategy to enhance the SRR performance, responding to abrupt and large traffic modifications with an effective routing adaptation. Volume anomalies may have an important impact on network performance, causing sudden situations of strong congestion. The early detection and localization of these anomalies allows to adjust routing as soon as possible, limiting their impact. Unfortunately, there are several shortcomings in current algorithms for anomaly detection in the TM that limit their usefulness in the practice: (1) most detection methods rely on highly tuned data-driven traffic models that are not stable in time, hence they are not appropriate for the task; (2) current detection methods present a lack of theoretical support on their optimality properties, making it almost impossible to compare their performances; (3) most approaches only treat the detection of the anomaly, but the anomaly localization and the application of countermeasures are still open problems; just detecting the anomaly does not solve the real problem.

1.1 Related Work

There is a large literature on routing optimization for uncertain traffic demands. Traditional algorithms rely on a small group of TMs to compute quasi-optimal routing configurations. An extreme case is presented in [6], where routing is optimized for a single estimated TM and it is then applied for daily routing. Traffic uncertainty is characterized by multiple TMs in [7], and different mechanisms to find optimal routes for these TMs are presented. Given the dynamic nature of current traffic demands, the traditional perspective may no longer be suitable for current scenario [8]. A different perspective is provided by Dynamic Load Balancing (DLB) algorithms: TeXCP [9] and MATE [10] both balance load in real-time, responding to instantaneous traffic variations. Their main goal is to avoid network congestion by adaptively balancing the load among fixed paths, based on measurements. DLB presents a desirable property, that of keeping routing adapted to traffic variations. However, as we shall see, these adaptive algorithms may present poor performance under significant and abrupt traffic changes. A third category of algorithms consists in Stable Robust Routing (SRR) techniques [3–5]. In [3], authors capture traffic variations by introducing a polyhedral set of demands, applying linear programming techniques to compute an optimal single routing configuration for all the traffic demands within this set. Oblivious Routing [4] also defines linear algorithms to optimize worst-case performance for different sizes of traffic uncertainty sets. The drawback of SRR is its inherent dependence on the definition of the uncertainty set: larger sets allow to handle more variable traffic demands, but at the cost of routing inefficiency; tighter sets produce more efficient routings, but are subject to poor performance guarantees. In [5], authors introduce an approach to deal with this trade off in the size of the uncertainty set, combining traditional algorithms with oblivious routing. Routing is optimized for expected

traffic, and bounds are provided for worst-case performance in the event of large traffic variations. Nevertheless, this approach proposes a single routing configuration as previous works do, losing the adaptability, and hence the performance efficiency of dynamic algorithms. We shall evidence that it is possible not only to ensure performance guarantees for unexpected events, but to obtain near-optimal routing configurations to deal with this traffic.

As regards network anomaly detection, the problem has been extensively studied. We just overview the most representative works for network-wide anomaly detection in the TM, using aggregated link traffic measurements as input, SNMP measurements from now on [2–12]. In [2], authors use Principal Components Analysis (PCA) and the sub-space technique to detect and localize network-wide anomalies in the TM, separating the SNMP measurements into anomalous and anomaly-free traffic representations. The PCA approach is a data-driven method which highly depends on the particular set of SNMP measurements under analysis, and as shown in [13], it must be highly tuned to provide accurate results, becoming impractical. [11] analyzes SNMP measurements using ARIMA modeling, Fourier transforms, wavelets, and PCA to model traffic evolution, detecting and locating anomalies as large deviations. Unfortunately, the technique presents a lack of theoretical results on its optimality properties (e.g. maximization of the probability to detect an anomaly with a bounded false alarm rate), a necessary condition to provide robust and easy-to-generalize results. [12] uses a Kalman-filtering approach to track the evolution of OD flows from SNMP measurements, detecting anomalies as large prediction errors. The method requires a long training phase where direct anomaly-free OD flow measurements are used to calibrate the underlying model. As we have recently shown [14], the assumed model has a particular structure that may require several periodical re-calibrations to provide reliable results, which makes it too costly to implement in the practice. Besides, the paper does not tackle the anomaly localization problem.

1.2 Contributions of the Paper

This work proposes both proactive and reactive complementary approaches to manage dynamic traffic demands, separately treating both sources of traffic variation. For *expected traffic variations*, we present a time varying extension of SRR that outperforms current single static-routing approach: Multi-Hour Robust Routing (MHRR). SRR may be costly: however, it is easy to control its cost by shrinking the uncertainty set. In MHRR, the uncertainty set is optimally divided into several sub-sets, considering the time direction for the partitioning. MHRR preserves the virtues of SRR, but changes the routing configuration during time, applying a SRR configuration for each period of the day. For the case of *unexpected traffic variations*, we present an optimal algorithm for volume anomaly detection and localization, permitting to identify strong traffic variations, locate their origins, and perform accurate routing changes. To overcome the stability problems of previous anomaly detection approaches, a novel linear, parsimonious, non data-driven traffic model is proposed. The model is used to filter the anomaly-free traffic from the SNMP measurements, providing residuals sensitive to anomalies. The

main contribution of the detection/localization algorithm relies on its well-established optimality properties, a fundamental feature generally absent in previous works.

Both proactive and reactive methods are combined into a novel approach to handle dynamic traffic demands: Reactive Robust Routing (RRR). Figure 1 presents a high level description of this approach. RRR uses MHRR to handle normal variations in traffic demands, and the detection/localization algorithm to deal with unexpected volume anomalies. RRR exploits the localization ability to deploy an adapted robust routing configuration after the anomalous traffic detection, reducing its impact on network performance during its prevalence. In addition, it also provides a simple yet effective method to automatically detect the end of the anomaly, regaining the MHRR configuration. Contrary to previous works in the field, this proposal optimizes routing in a robust and adaptive fashion for every possible traffic demand, and not only for the common-case or expected traffic. A key feature of RRR relies on the fact that the whole routing modification algorithm is completely automatic, an interesting property that simplifies network operation. The proposed algorithms are validated using real traffic from three different backbone networks: the Internet2 Abilene backbone network, the European GEANT network, and a private international Tier-2 network. This work represents a continuation of our previous works on robust routing [8, 15] and anomaly detection [16].

The remainder of this paper is organized as follows. In Sect. 2, the basic concepts of the robust routing approach are recalled. Section 3 presents the theoretical background and empirical evaluation of MHRR. The linear parsimonious TM model is introduced and validated in Sect. 4. Section 5 describes the anomaly detection/localization algorithm and presents an empirical validation using real traffic. Section 6 presents the Reactive Robust Routing approach, showing the automatic interaction between the proactive and the reactive components through complete real and simulated anomaly scenarios. In this section we also propose a load-balancing

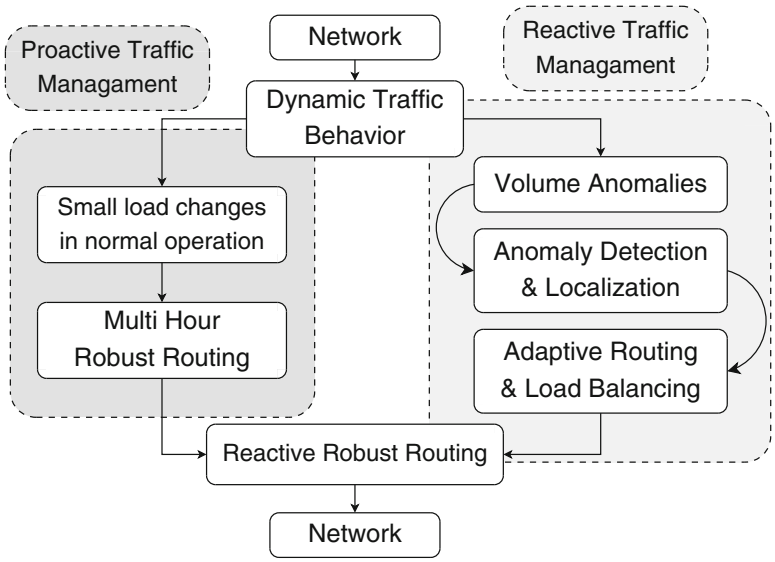


Fig. 1 High level description of the Reactive Robust Routing approach

extension for RRR nicknamed Reactive Robust Load Balancing (RRLB), in which end-to-end paths remain fixed and only routing fractions are modified. We further compare this extension against traditional DLB methods, showing the impressive stability of RRLB when faced with volume anomalies. Finally, Sect. 7 concludes this work.

2 Robust Routing

Let us consider a network topology defined by a set of n nodes and a set of r links $L = \{1, ..., r\}$ with capacities in $C = \{c_1, c_2, ..., c_r\}$. The TM $\mathbf{d} = \{d(i, j)\}$ denotes the traffic flow between every node i and node $j (i \neq j)$ of the network. This matrix is arranged as a column vector, $\mathbf{d} = \{d(k)_{,\ k=1..m}\}$, where $d(k)$ represents the Origin-Destination traffic flow transmitted by each OD pair of nodes k, and $m = n\ .\ (n - 1)$ is the number of OD pairs. Let $N = \{\mathrm{OD}_1, ..., \mathrm{OD}_m\}$ be the set of m OD pairs. Let $y(l)$ be the total aggregated traffic at link l in a certain period of time. This data is available from routers management information and it is periodically collected via the well-known SNMP protocol. Traffic demands and links traffic are related through the routing matrix R, a $r \times m$ matrix $R = \{r_{l,k}\}$ where $0 \leqslant r_{l,k} \leqslant 1$ represents the fraction of OD flow k routed through link l: $\mathbf{y} = R\ .\ \mathbf{d}$, with $\mathbf{y} = \{y(l)_{,l=1..r}\}$. Routing optimization depends on the underlying data transport mechanism; in this case, the focus is on path-based routing such as MPLS. This optimization consists of minimizing certain performance index associated with traffic demands and network topology. Throughout this work, the Maximum Link Utilization (MLU) is considered as the routing performance criterion. For a given routing matrix R and a traffic demand $\mathbf{d}$, the MLU $u_{\max}$ is defined as the maximum of the ratio between link traffic and link capacity (1):

$$u_{\max}(C, \mathbf{d}, R) = \max_{l \in \{1...r\}} \sum_{k=1}^{m} \frac{r_{l,k} \cdot d(k)}{c_l} = \max_{l \in \{1...r\}} \frac{y(l)}{c_l} \tag{1}$$

While this criterion does not represent a direct measure of Quality of Service (QoS), overloaded links tend to cause QoS degradation (e.g. larger delays and packet losses, throughput reduction, etc.), so MLU represents a reasonable measure of network performance. There are many other performance indexes that could be used instead of MLU, like end-to-end path delay or mean link utilization; setting the focus too strictly on the MLU can often lead to longer average traffic paths and thus adversely affect the mean. However, the MLU is by far the most commonly applied criterion as it represents an easy to understand performance measure, so it will be the adopted one. Let $P(k)$ be the set of possible paths for OD flow k. Let $r_{p,k}$ be the proportion of traffic demand $d(k)$ that flows through path $p \in P(k)$, $0 \leqslant r_{p,k} \leqslant 1$. Link routing fractions $r_{l,k}$ can be directly computed from path routing fractions $r_{p,k}$:

$$r_{l,k} = \sum_{p \in P(k),\, p \ni l} r_{p,k} \qquad \forall k \in N,\ \forall l \in L \tag{2}$$

$$
\begin{aligned}
&\text{minimize} \quad u_{\max} \\
&\text{subject to:} \\
&\sum_{p \in P(k)} r_{p,k} = 1 \qquad \forall k \in N \\
&\sum_{p \in P(k),\, p \ni l} r_{p,k} = r_{l,k} \qquad \forall k \in N,\ \forall l \in L \\
&\sum_{k \in N} r_{l,k}.d(k) \leqslant u_{\max}.c_l \qquad \forall l \in L,\ \forall \mathbf{d} \in D \\
&r_{p,k}, r_{l,k} \geqslant 0 \qquad \forall l \in L, \forall p \in P(k),\ \forall k \in N \\
&u_{\max} \leqslant 1
\end{aligned}
\tag{3}
$$

Routing optimization for traffic demand $\mathbf{d}$ can be easily performed when this demand is perfectly known. However, in the practice, $\mathbf{d}$ is unknown and all we may know for sure is that it belongs to a certain bounded set D in which it can vary. This set represents the *uncertainty* in the value of $\mathbf{d}$, and so it is usually known as the *uncertainty set*. The set D can be defined in different ways, depending on the available data: SNMP measurements and historical routing, a set of previously measured TMs, time series of TMs $\mathbf{d}_t$, etc. Authors in [3] define this set as a *polytope*, based on the intersection of several half-spaces that results from linear constraints imposed to traffic demands. As a practical example, we shall define a simple uncertainty set D, based on the routing matrix R and the busy-hour links traffic $\mathbf{y}_{\text{busy}}$:

$$D = \left\{ \mathbf{d} \in \mathbb{R}^m, R.\mathbf{d} \leqslant \mathbf{y}_{\text{busy}},\ \mathbf{d} \geqslant 0 \right\} \tag{4}$$

Figure 2a depicts the obtained polytope D, based on the convex intersection of r half-spaces $\mathbf{r}_i \cdot \mathbf{d} \leqslant y_{\text{busy}}(i)$, $\forall i \in \{1 \ldots r\}$, where $\mathbf{r}_i$ stands for the i-th row of R.

The Robust Routing Optimization Problem (RROP) defined in (3) consists of minimizing $u_{\max}$ for all the traffic demands within the uncertainty set D. This linear system can be efficiently solved by linear programming techniques, applying a combined columns and constraints generation method [3]. RROP represents a worst-case optimization for all the traffic demands inside D, hence it provides performance guarantees $\forall \mathbf{d} \in D : u_{\max}(C, \mathbf{d}, R_{\text{robust}}) \leqslant u^*_{\max},\ \forall \mathbf{d} \in D$, where $u^*_{\max}$ and $R_{\text{robust}} = \left\{ r^*_{l,k} \right\}$ are the solution to (3). Note that using (4) as uncertainty set has

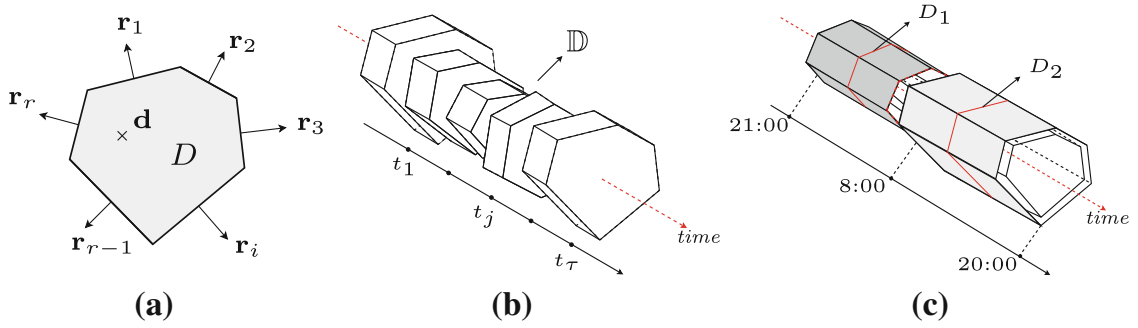


Fig. 2 **a** The set D as a polytope, **b** daily polytope $\mathbb{D}$, **c** time partitioning of $\mathbb{D}$

a major advantage: routing optimization can be performed from easily available SNMP measurements, without even measuring any TM at all. In a traditional robust routing application, the obtained routing configuration R_{robust} is applied during long periods of time, usually in a daily routing basis. In this sense, the robust routing approach is usually referred to as Stable Robust Routing (SRR).

3 Multi-Hour Robust Routing

The advantages of SRR with respect to traditional routing approaches are presented in [8]; to sum up, SRR offers stability guarantees against traffic uncertainty and normal traffic variations at a reasonable cost. However, considering a single routing configuration for long periods of time is not a cost-effective solution and results in sub-optimal performance. We therefore propose an approach to shrink and adapt the uncertainty set that outperforms SRR. Based on rough knowledge of traffic variations, basically considering expected traffic behavior, the uncertainty set is optimally divided in the direction of the time, producing several sub-sets; a multi-hour routing configuration is then built, considering a single SRR configuration for each of these sub-sets.

Daily traffic variations can be seen as a time variation of the uncertainty set. At each time slot t_j, the routing matrix R and the SNMP measurements $\mathbf{y}_{t\_j}$ define an uncertainty set $D(t_j) = \left\{\mathbf{d} \in \mathbb{R}^m, R.\mathbf{d} \leqslant \mathbf{y}_{t_j}, \mathbf{d} \geqslant 0\right\}$. The slotted time comes from the fact that SNMP measurements $\mathbf{y}_{t_j}$ are collected at discrete time intervals t_j, typically every 5 or 10 mins. The union of several uncertainty sets along contiguous time slots $t_1, .., t_j, .., t_\tau$ defines a *temporal uncertainty set* $\mathbb{D} = \left\{\mathbf{x} = \left\{\mathbf{d}_{t_j}, t_j\right\} \in \mathbb{R}^{m+1}, \mathbf{d}_{t_j} \in D(t_j), t_j \in [t_1..t_\tau]\right\}$. Figure 2b explains this idea. Assuming that this set is a union of polytopes, [17] provides a theoretical study of the optimal partitioning of $\mathbb{D}$, using a partitioning hyper plane. In particular, it proves that this is a NP-hard problem, except for the case where a partitioning direction is previously fixed. In such a case, the author presents a simple algorithm to approximately solve (5) in polynomial-time, using a generalization of a simple dichotomy methodology.

A partitioning hyper plane is defined by its direction vector α and a value β:α. $\mathbf{x} = \beta$. In MHRR there is a particular direction for partitioning: the *time direction*. In this case, $\alpha = [0, .., 0, 1]$ and $\beta = t$. Given β_1 and β_{h+1}, we define $h - 1$ hyperplanes at times $\boldsymbol{\beta} = \{\beta_2, .., \beta_h\}$. Let D_i be the convex hull of the union $\left\{\cup_{\beta_i \leqslant t_j \leqslant \beta_{i+1}} D(t_j)\right\}, \forall i = 1, ., h$, see Fig. 2c. MHRR consists in computing the optimal times $\boldsymbol{\beta}^*$ when routing should be modified, in order to minimize the worst case of MLU in $\mathbb{D}$; $\boldsymbol{\beta}^*$ is the solution to (5), where $u^*_{\max}(D_i)$ is the solution to (3) for polytope D_i.

$$\boldsymbol{\beta}^*(\mathbb{D}) = \arg\min_{\boldsymbol{\beta}} \left\{ \max_{i=1..h} u^*_{\max}(D_i) \right\} \tag{5}$$

Finally, a single SRR configuration is computed for each time interval $\left[\beta^*_i, \beta^*_{i+1}\right], \forall i = 1, ., h$. The interesting issue in our proposal is that we provide an

objective means of computing an optimal multi-hour routing design, maintaining the robustness of the SRR approach. The optimality property of our approach lies on the computation of the ideal times $\boldsymbol{\beta}^*$ to switch routing. Traditional methods used in the design of multi-hour routing configurations are rather simplistic, relying on a couple of TMs to optimize different routing configurations [18].

MHRR presents a trade-off between performance and routing stability. The more intervals, the more adapted the routing becomes. However, the number of intervals should be bounded as many routing changes may lead to instabilities and performance degradation. In a general case, 2 sub-sets are enough to handle the usual daily variation [8]. Additionally, and as we shall explain in Sect. 6.3, it is possible to modify just the routing fractions $r_{p,k}$ and keep the same paths $P(k)$ used in each SRR, avoiding routing reconfigurations. In other words, it is possible to do Multi-Hour Load Balancing instead of MHRR.

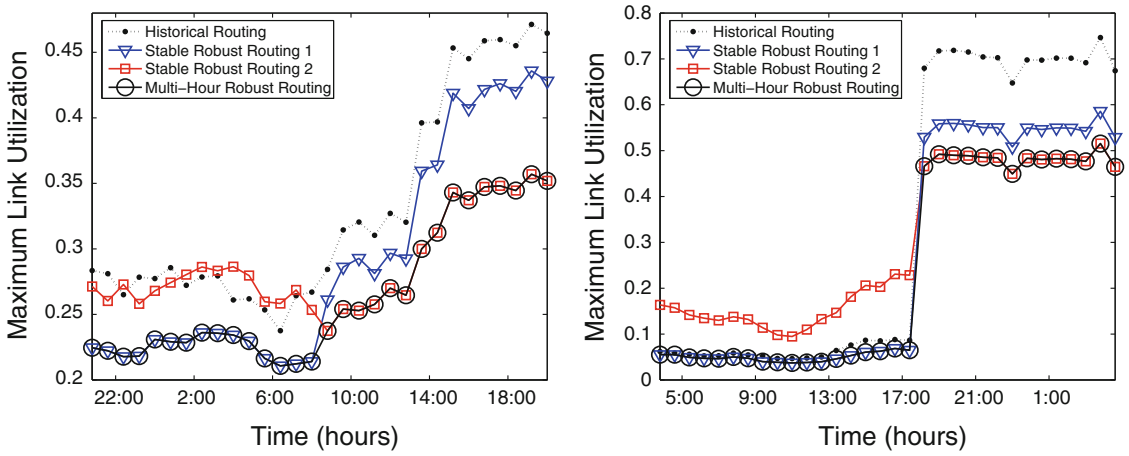
3.1 MHRR Evaluation

The SRR and the MHRR approaches are compared in Abilene, an Internet2 backbone network. Abilene consists of 12 router-level nodes and 30 links. The used network topology and traffic demands are available at [19]. Traffic data consists of 6 months of TMs collected via Netflow from the Abilene Observatory [20] in 2004. As the measured traffic demands do not significantly load the network, we have rescaled them by multiplying all their entries by a constant factor. Since the Abilene network spans different time zones, the time variation of the polytope is not a simple homothety: in fact, there is no strong traffic synchronization between links. In this sense, a routing scheme modification during the day improves routing performance. Let R_o be the historical routing matrix used in Abilene, which is not necessarily optimal. This routing matrix is also available at [19]. A single time partitioning is considered (i.e. 2 routing intervals), $\beta_1 = 21{:}00$, $\beta_2 = \beta^*$, and $\beta_3 = 20{:}00$, where β^* is the solution to (5). The smallest polytope that includes all possible realizations over that period is computed for each time interval:

$$D_1 = \{\mathbf{d} \in \mathbb{R}^m, R_o.\mathbf{d} \leqslant \mathbf{y}_1,\ \mathbf{d} \geqslant 0\} \tag{6}$$

$$D_2 = \{\mathbf{d} \in \mathbb{R}^m, R_o.\mathbf{d} \leqslant \mathbf{y}_2,\ \mathbf{d} \geqslant 0\} \tag{7}$$

where $\mathbf{y}_1 = \mathbf{y}^{\max}_{21:00-\beta^*}$ and $\mathbf{y}_2 = \mathbf{y}^{\max}_{\beta^*-20:00}$ include the maximum values for each link in the corresponding period. In this way, D_1 includes all the traffic demands between 21:00 and β^*, and D_2 between β^* and 20:00, see Fig. 2c. For each polytope, a SRR configuration is computed, R^1_{robust} and R^2_{robust}. In order to compare the stable and the multi-hour approaches, both routing configurations are applied during the whole evaluation period. The routing performance obtained with R_o is also included, which corresponds to the dotted line with label *Historical Routing* in Fig. 3. Figure 3a compares the MLU obtained by these two SRR configurations. Polytope D_1 is better suited for smaller loads, so R^1_{robust} performs better during the first half of the day, when network load is lower. However, when traffic increases, demands that do not belong to D_1 produce higher link utilizations than those obtained with R^2_{robust}. MHRR consists of computing the optimal time when routing must be changed,



(a) Expected daily behavior **(b)** Anomalous unexpected event

Fig. 3 Routing performance, Stable versus Multi-Hour Robust Routing. **a** Traffic corresponds to TMs from the 1st week of the dataset, starting on the 01/03/04. **b** TMs from the 23th week, starting on the 28/08/04

using the corresponding routing configuration depending on the time of the day: R^1_{robust} before β^* and R^2_{robust} after. In this evaluation, the computed value of $\beta^* \approx$8:00. MHRR presents a performance improvement of 16% w.r.t. SRR before β^*, reaching a near 20% of over-efficiency after β^*. Traffic demands that drastically change due to a volume anomaly are considered as a second case study. Figure 3b presents an huge abrupt change in MLU at time 18:00. In this case, we shall assume that the change is known in advance; note that in the general case it is not possible to predict such abrupt variations. The optimal time for switching routing is $\beta^* \approx$ 18:00. MHRR definitely outperforms SRR in this traffic scenario, providing a MLU between 10 and 60% smaller during the evaluation period.

4 Dealing with Unexpected Events

The proposed MHRR approach offers a robust and efficient routing configuration, given a rough knowledge of the temporal uncertainty set. However, in the presence of volume anomalies it is no longer possible to apply MHRR; the optimal division proposed in (5) cannot be done if $\mathbb{D}$ presents strong and unknown variations. A reactive approach is proposed for those cases, based on the detection and localization of these volume anomalies. To avoid costly and difficult to perform direct OD flow measurements, the detection and localization algorithm uses SNMP measurements $\mathbf{y}_t$ as input data. Additional flow-based technology is necessary to network-wide collect and process direct OD flow measurements [21], so different traffic models have been developed using SNMP measurements and routing information to *reconstruct* OD flows. This reconstruction represents an ill-posed problem, because the number of unknown OD flows is much larger than the number of links [21]; briefly, it is not possible to compute $\mathbf{d}_t = \{d_t(1), ..., d_t(m)\}$ from $\mathbf{y}_t = R \,.\, \mathbf{d}_t$, because $r \ll m$. To overcome this problem, a novel parsimonious, linear model for normal-operation traffic is proposed next. In this work, this traffic model

is used to remove the anomaly-free traffic from the anomaly detection problem. However, the model can be used to solve other problems, like the well-known TM estimation problem among others.

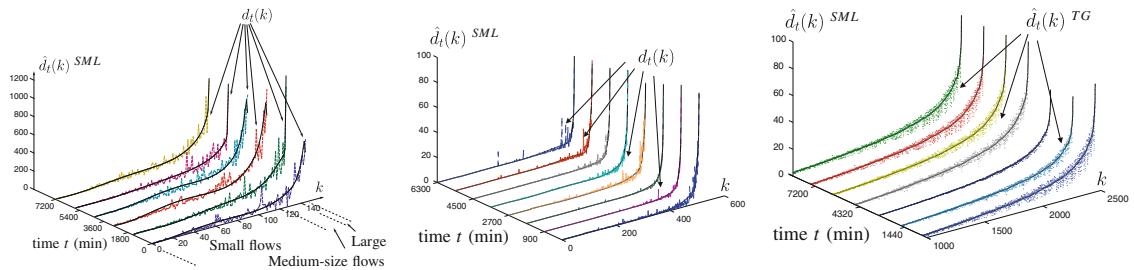
4.1 Stochastic Traffic Model for Anomaly Detection

We assume that the stochastic process of the TM $\mathbf{d}_t$ obeys the following linear expression:

$$\mathbf{d}_t = \boldsymbol{\lambda}_t + \boldsymbol{\xi}_t \tag{8}$$

where $\boldsymbol{\lambda}_t \in \mathbb{R}^m$ is the mean traffic demand and $\boldsymbol{\xi}_t$ is a white Gaussian noise with covariance matrix $\Sigma = \mathrm{diag}\left(\sigma_1^2, \ldots, \sigma_m^2\right)$. The process $\boldsymbol{\lambda}_t$ represents the temporal evolution of the mean TM, which can be correctly modeled in the absence of anomalies. The white Gaussian noise $\boldsymbol{\xi}_t$ models the natural variability of the TM together with the modeling errors. In order to describe $\boldsymbol{\lambda}_t$ with a small number of coefficients, a key feature of the TM is employed: its spatial stationarity. Many classical TM models make use of this assumption, like the very well-known gravity model [22, 23]. The other key observation for this model is the *mice and elephants phenomenon*: a small percentage of OD flows contribute to a large proportion of the total traffic in the TM [22]. The existence of such dominant flows together with the spatial stationarity of flows makes it reasonable to assume that, in the absence of an anomaly, the largest OD flows in a network remain the largest, and the smallest flows remain the smallest during long periods of time; this assumption is confirmed in the empirical validation of the model, at least for several days, see Sect. 4.3. Therefore, it seems logical to accept that the order of increasing OD flows w.r.t. their traffic volume remains stable in time. The sorted OD flows can be interpreted as a discrete non-decreasing signal with certain smoothness. The curve obtained by interpolating this signal is assumed to be a continuous curve, hence it can be parameterized by a polynomial splines approximation.

Figure 4 shows the anomaly-free OD flows for three different operational networks, sorted in the increasing order of their volume of traffic, for different time instants t. The dashed lines depict the value of each sorted OD flow $d_t(k)$, $k = 1..m$,



(a) Abilene Network **(b)** GEANT Network **(c)** Tier-2 ISP Network

Fig. 4 Approximation of real OD flows (*dashed lines*) by the spline-based model (*full lines*) in 3 operational networks. $\hat{d}_t(k)^{TG}$ is the estimated OD flow k using the tomogravity estimation method, introduced in [23]. TMs come from the week starting on the **a** 01/03/04, **b** 01/06/05, and **c** 18/04/05, respectively

the full lines represent the polynomial approximation of the sorted flows. In order to appreciate the time stability of this approximation, the curves are plotted for several consecutive days. Given the shape of the curve formed by the sorted OD flows, a cubic splines approximation is applied; basic definitions and results on polynomial splines can be found in [24]. A discrete spline basis is designed, discretizing the continuous splines according to m points uniformly chosen in the interval [1;m] and rearranging them according to the OD flows sorting order. The obtained linear parsimonious model for the anomaly-free traffic demand can be expressed as:

$$\mathbf{d}_t = S\boldsymbol{\mu}_t + \boldsymbol{\xi}_t \tag{9}$$

where $S = \{\mathbf{s}(i),\ i = 1..q\}$ is a $m \times q$ known matrix with a small number of columns w.r.t. the number of OD flows, i.e., $q \ll m$. The vectors $\mathbf{s}(i)$, which correspond to the rearranged discrete splines, form a set of known basis vectors describing the spatial distribution of traffic. The vector $\boldsymbol{\mu}_t = \{\mu_t(1)\ldots\mu_t(q)\}^T$ is the unknown time varying parameter that describes the OD flow intensity distribution w.r.t. the set of vectors $\mathbf{s}(i)$. The model for the anomaly-free links traffic is given by:

$$\mathbf{y}_t = G\boldsymbol{\mu}_t + \boldsymbol{\zeta}_t, \tag{10}$$

where $G = RS$ and $\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \Phi)$, with $\Phi = R\Sigma R^T$. The computation of the rank of G is not simple since it depends on R. In the practice, since the number of columns of G is very small, the product RS and its rank can be computed very fast. Therefore, it will be assumed that G is full column rank. To simplify notation and computations, the *whitened* measurements vector is introduced:

$$\mathbf{z}_t = \Phi^{-\frac{1}{2}}\mathbf{y}_t = H\boldsymbol{\mu}_t + \epsilon_t, \tag{11}$$

where $H = \Phi^{-\frac{1}{2}}G$ and $\epsilon_t \sim \mathcal{N}(0, I_r)$ (I_r is the $r \times r$ identity matrix). The purpose of this transformation is simply to whiten the Gaussian noise. Finally, the covariance matrix Σ is unknown. The solution consists in computing a simple empirical estimate $\widehat{\Sigma}$ from a few anomaly-free measurements.

4.2 Validation of the Model: The Dataset

The validation of the proposed traffic model is conducted using real data from two operational networks: the Abilene network previously presented, and the European GEANT research network. GEANT traffic data consists of 15' sampled TMs, built from IGP and BGP routing information and Netflow data, available at the TOTEM website [25]. In the following evaluations, we assume that traffic demands $\mathbf{d}_t$ are unknown and just consider the link load values $\mathbf{y}_t$ as the known data.

In order to verify the stability properties of the model, two sets of measurements are used: the first one, the *learning* anomaly-free dataset, is composed of one hour of anomaly-free SNMP measurements (3 h in GEANT, due to the different sampling rates) and it is used to construct the spline basis S; the second one, the *testing* dataset, is composed of 672 SNMP measurements and it is used to validate the model. Let T_{learning} and T_{testing} be the set of time indexes associated with SNMP

measurements from the learning and the testings datasets, respectively. The learning dataset is measured immediately before the testing dataset.

4.3 Numerical Validation of the Model

In order to validate the splines model, we shall consider an estimate of the real traffic demands and test its accuracy. We shall construct a Maximum-Likelihood (ML) estimate for $\mathbf{d}_t$. The statistical properties of the ML estimate are very well-known, and thus it represents an excellent estimation approach. Since the traffic model is a Gaussian model, the Spline-based Maximum Likelihood (SML) estimate $\hat{\mathbf{d}}_t^{SML}$ corresponds to a simple least-squares estimate:

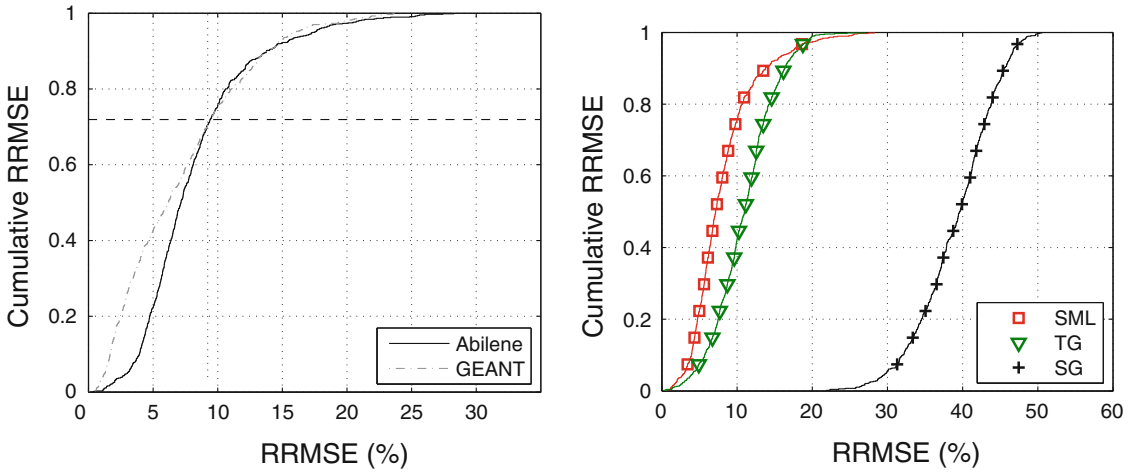
$$\hat{\mathbf{d}}_t^{SML} = \left(S\left(H^T H\right)^{-1} H^T \Phi^{-\frac{1}{2}}\right)\mathbf{y}_t \tag{12}$$

The Spline-based model is computed using the learning dataset, following these steps: (1) a tomogravity estimate $\hat{d}_t^{TG}(k)$ [23] is computed for all OD flows k and all $t \in T_{\text{learning}}$, (2) the mean OD flow values $\bar{d}^{TG}(k)$ are computed: $\bar{d}^{TG}(k) = \frac{1}{\#(T_{\text{learning}})} \sum_t \hat{d}_t^{TG}(k)$, and (3) they are sorted in ascending order to obtain a rough estimate of the OD flows traffic volume. The Spline-based model is designed with cubic splines and 2 knots, representing small, medium-size, and large OD flows. The estimates $\hat{d}_t^{TG}(k)$ and mean values $\bar{d}^{TG}(k)$ are also used to compute an estimate $\hat{\sigma}_k^2$ of σ_k^2, which leads to an estimate $\hat{\Phi}$ of Φ.

As a global indication of the accuracy of the SML estimate, and to test its performance against some other well-known traffic estimation methods, we shall use the relative Root Mean Squared Error (RRMSE) for each time t of the testing dataset:

$$\text{RRMSE}(t) = \frac{\sqrt{\sum_{k=1}^{m} \left(d_t(k) - \hat{d}_t^{\text{label}}(k)\right)^2}}{\sqrt{\sum_{k=1}^{m} d_t(k)^2}}, \ \forall t \in T_{\text{testing}} \tag{13}$$

where $d_t(k)$ is the true traffic volume of the anomaly-free OD flow k at time t and $\hat{d}_t^{\text{label}}(k)$ denotes the corresponding estimate for the method entitled 'label'. The RRMSE provides at each time t a summary of the relative estimation error for the TM. Figure 5a depicts the CDF of the RRMSE errors for the 672 TMs in Abilene and GEANT, showing that in more than 70% of the time indexes, estimation errors are below 10%. A deeper study of the RRMSE shows that in most cases, large RRMSE values correspond to large relative errors in the smallest-volume OD flows, which are well known to be hard to estimate [23, 26]. Note however that small OD flows have little impact on traffic engineering tasks, hence are generally less important to estimate. The mean values of the RRMSE for the evaluation period are 8.14% in Abilene and 7.04% in GEANT. Methods proposed in the literature as "accurate" estimates present relative errors that vary between 5 and 15% [26], so obtained results are satisfactory.



(a) CDF - RRMSE for the SML estimates. **(b)** CDF - RRMSE in Abilene.

Fig. 5 Comparison between the SG, TG and SML estimates for 672 anomaly-free TMs from the week starting on the 01/03/04 in Abilene, and on the 01/06/05 in GEANT

Figure 5b compares in Abilene the performance of the SML estimate with two very well-known methods for TM estimation: the simple gravity and the tomogravity estimates [23], with labels 'SG' and 'TG', respectively. The RRMSE corresponding to the tomogravity estimate TG is quite close to the error produced with our model. However, the SML estimate presents a major advantage w.r.t. the TG estimate: as it was previously said, the ML estimate presents well established statistical properties, which is not the case for the TG estimate. The SML estimate is *asymptotically optimal*, i.e., it is asymptotically unbiased and efficient. Moreover, the Spline-based model can be used in order to design anomaly detection algorithms with optimality properties, which is not the case for the tomogravity estimate.

As a final validation, the Gaussian assumption of the model is studied. The *residuals* of measurements are analyzed, i.e. the obtained traffic after filtering the "regular" part $H\boldsymbol{\mu}_t$. The residuals are obtained by projection of the whitened measurements vector $\mathbf{z}_t = \Phi^{-\frac{1}{2}}\mathbf{y}_t$ onto the left null space of H, obtaining a residuals vector $\mathbf{u}_t = W\mathbf{z}_t \sim \mathcal{N}(0, I_{r-q})$, see the Appendix for computation details about the *rejection matrix* W. A simple Kolmogorov-Smirnov test at the level 5% accepts the Gaussian hypothesis for 662 of the 672 anomaly-free measurements, representing an acceptation ratio of 98.5%. This confirms the Gaussian assumption of the model.

5 Anomaly Detection and Localization

Once the traffic model has been introduced and validated, the reactive component of our proposal is presented. The goal of the proposed method is to detect and localize an additive change $\theta\ \mathbf{r}_j$ in a series of SNMP measurements, where $\mathbf{r}_j$ is the j-st column of the routing matrix R, and θ is the intensity of the anomaly; this corresponds to a change θ in OD flow j. In this work we consider the same simplifying hypothesis as in [2], considering only "localized" anomalies, namely anomalies in a single OD flow at a time. As it is shown in [2, 27], this hypothesis is

extensively verified in both the Abilene network and the Sprint-Europe network, the European backbone of a US tier-1 ISP. For example, [27] shows that more than 85% of the volume anomalies detected in Abilene correspond to single-OD flow anomalies. The simplifying hypothesis is therefore adequate in real-traffic scenarios.

We propose an optimal sequential algorithm to detect abrupt changes in traffic demands. Sequential approaches are used to minimize the number of observations needed to detect an anomaly after its occurrence. The sequential algorithm not only detects an abrupt change in the TM, but also localizes the change, i.e. it identifies the OD flow in which the change has occurred. This detection/localization algorithm is optimal in the sense that it minimizes the maximum mean detection/localization delay for a given maximum probability of false localization η and minimum mean time before a false alarm υ, a usual measure of the false alarm rate. The localization of the anomalous traffic is possible since an anomaly in a given OD flow typically spans multiple links, and has a particular *signature* as a result of the routing process.

The detection/localization of a volume anomaly that occurs at an unknown time t_0 can be treated as a hypothesis testing problem, where the null hypothesis $\mathcal{H}_0 = \{\text{OD flows are anomaly-free}\}(t_0 = +\infty)$ is tested against m alternatives $\mathcal{H}^j_{t_0} = \{$the j-th OD flow presents an anomalous additional amount of traffic θ from time $t_0\}$, $j = 1..m$. The algorithm has to compute the alarm time T_r at which a v_r-type change $\in \{1, 2, ..., m\}$ is detected and localized, based on SNMP measurements $\mathbf{y}_1$, $\mathbf{y}_2$, ... The hypothesis testing can be written as:

$$\mathcal{H}_0 : \mathbf{z}_t \sim \mathcal{N}(H\boldsymbol{\mu}_t, I_r),\ t = 1, 2, \ldots, \tag{14}$$

$$\mathcal{H}^j_{t_0} : \begin{cases} \mathbf{z}_t \sim \mathcal{N}(H\boldsymbol{\mu}_t, I_r),\ t = 1, \ldots, t_0 - 1, \\ \mathbf{z}_t \sim \mathcal{N}(H\boldsymbol{\mu}_t + \theta\Phi^{-\frac{1}{2}}\mathbf{r}_j, I_r), \\ \quad \theta_{\text{lower}} \leqslant |\theta| \leqslant \theta_{\text{upper}},\ t = t_0, t_0 + 1, \ldots \end{cases} \tag{15}$$

where $\mathbf{z}_t$ is the whitened measurements vector and $0 < \theta_{\text{lower}} < \theta_{\text{upper}} < +\infty$ are some known bounds on the change intensity of the j-th OD flow that are introduced for technical reasons, but can be arbitrarily chosen. Hypothesis (15) can be rewritten by filtering the anomaly-free traffic (see the Appendix):

$$\mathcal{H}^j_{t_0} : \begin{cases} \mathbf{u}_t \sim \mathcal{N}(0, I_{r-q}),\ t = 1, \ldots, t_0 - 1, \\ \mathbf{u}_t \sim \mathcal{N}(\theta\mathbf{v}_j, I_{r-q}), \\ \quad \theta_{\text{lower}} \leqslant |\theta| \leqslant \theta_{\text{upper}},\ t = t_0, t_0 + 1, \ldots \end{cases} \tag{16}$$

where $\mathbf{u}_t = W\mathbf{z}_t$ are the residuals obtained from $\mathbf{z}_t$ after filtering the anomaly-free traffic, $\mathbf{v}_j = W\Phi^{-\frac{1}{2}}\mathbf{r}_j$ is a known non-null vector and the matrix W is the linear rejector that eliminates the anomaly-free traffic (see the Appendix). The vector $\mathbf{v}_j$ corresponds to the signature in the residuals of a change in OD flow j, specified by the column $\mathbf{r}_j$ of R.

The recursive algorithm proposed in [28] perfectly fits this detection and localization problem, with one ideal feature, that of minimizing the number of samples needed to decide between the hypotheses with bounded false localization and false alarm rates. The algorithm produces two outputs: a stopping time T_r that

represents the time when the anomaly is detected and a decision v_r about which OD flow carries the anomaly:

$$T_r = \min_{1 \leqslant k \leqslant m} \{T_r(k)\}, \quad v_r = \arg \min_{1 \leqslant k \leqslant m} \{T_r(k)\} \tag{17}$$

$$T_r(k) = \inf \left\{ t \geqslant 1 : \min_{0 \leqslant j \neq k \leqslant m} [g_t(k,j) - h_{k,j}] \geqslant 0 \right\}, \; k = 1, \ldots, m \tag{18}$$

with $g_t(k, j) = g_t(k, 0) - g_t(j, 0)$. At each time t, the recursive functions $g_t(k,0)$ give an idea of the difference between the value of traffic residuals under the hypothesis of normal operation and the hypothesis of anomaly. Each recursive function $g_t(k,0)$ is defined as:

$$g_t(k,0) = (g_{t-1}(k,0) + u_t(k,0))^+ \tag{19}$$

$$u_t(k,0) = \log \frac{f_k(\mathbf{u}_t)}{f_0(\mathbf{u}_t)} \tag{20}$$

where $(x)^+ = \max(x, 0)$, $g_0(k, 0) = 0$ for every $1 \leqslant k \leqslant m$, and $g_t(0,0) = 0$ for all t. The function f_0 represents the probability density function of residuals in normal-operation conditions. The function f_k is the probability density function of residuals $\mathbf{u}_{t_0}, \mathbf{u}_{t_0+1}$, ... after a change of type k. The thresholds $h_{k,j}$ are chosen by the following formula:

$$h_{k,j} = \begin{cases} h_\mathrm{d} \text{ if } 1 \leqslant k \leqslant m \text{ and } j = 0 \\ h_\mathrm{i} \text{ if } 1 \leqslant k, j \leqslant m \text{ and } j \neq k \end{cases} \tag{21}$$

where h_d and h_i are the detection and localization thresholds. $T_r(k)$ is the first time when one of the recursive functions $g_t(k, j)$ exceeds the thresholds $h_{k,j}$. The detection time T_r corresponds to the earliest off all the times $T_r(k)$, $1 \leqslant k \leqslant m$. The detected anomaly is declared in OD flow j if the earliest of all these times was $T_r(j)$. The algorithm is asymptotically optimal, i.e. it reaches the lower bound of the maximum mean delay for detection/localization [28], given bounds in the mean time between false alarms υ and in the probability of false localization η:

$$\mathbb{E}_0(T_r) \geq \upsilon \tag{22}$$

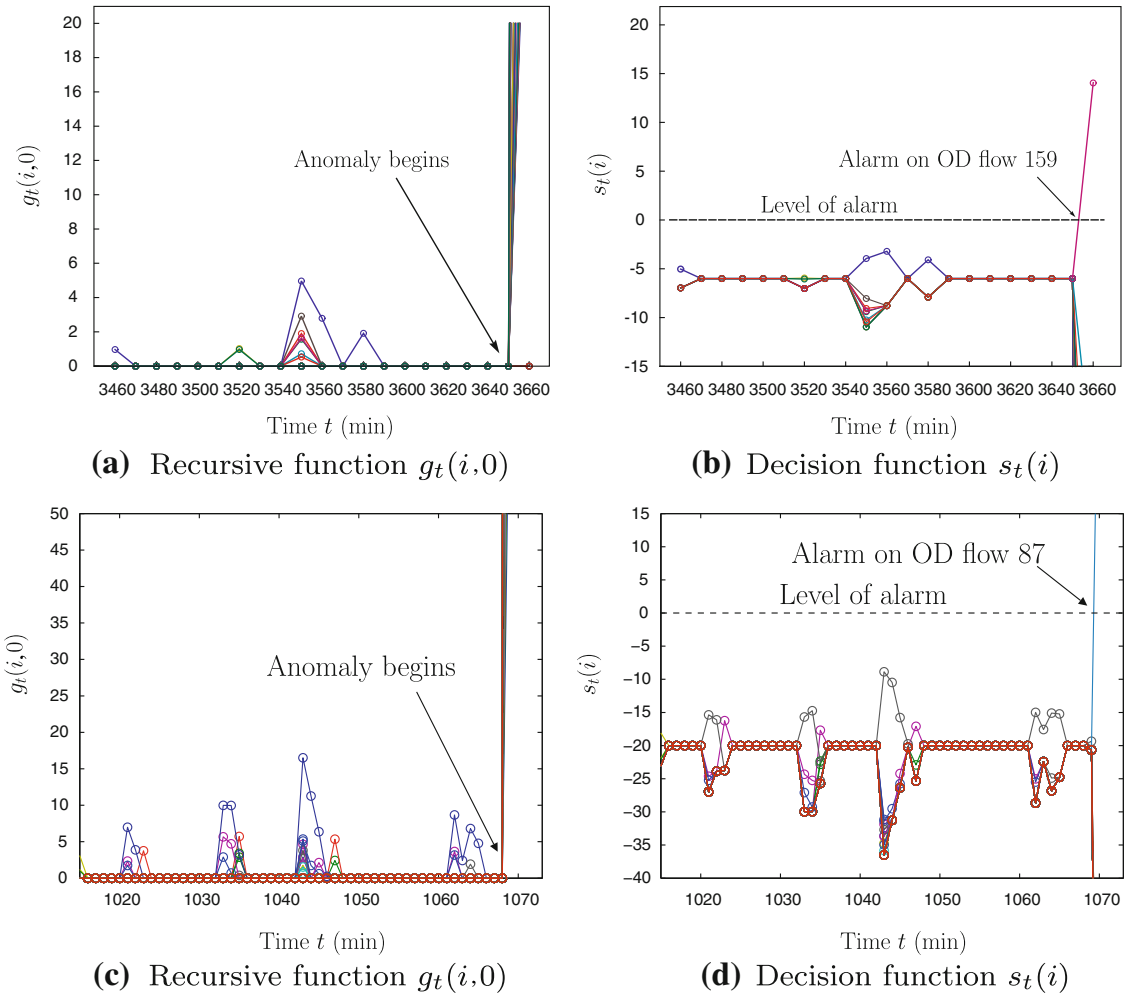
$$\mathrm{Pr}_{t_0}^k(v_r = j | T_r \geq t_0) \leq \eta, \tag{23}$$

$\mathbb{E}_0(\cdot)$ denotes the expectation when all the measurements have the same probability density function f_0 (anomaly-free behavior) and $\mathrm{Pr}_{t_0}^k$ corresponds to the probability (with density function f_k) that the sequential test declares the final decision j whereas the true change type is $k \neq j$. The choice of the detection and localization thresholds h_d and h_i is discussed in [29].

5.1 Numerical Validation of the Anomaly Detection/Localization Algorithm

We demonstrate the ability of the sequential algorithm to detect and localize a volume anomaly from SNMP measurements in two different networks: a large Tier-

2 network (50 nodes, 168 links and 2,450 non-zero OD flows, sampled at a 10' rate) and Abilene. Figure 6a,b show the typical realization of the decision functions $g_t(i,0)$ and $s_t(i) = \min 0 \leqslant k \neq i \leqslant m \ [g_t(i,k) - h_{i,k}]$ in the Tier-2 network. Functions $s_t(i)$ are used to "monitor" the OD flows; when $s_t(i)$ exceeds the threshold 0, OD flow i is declared anomalous. The anomaly begins at time 3,660. Note that after this time, several decision functions $g_t(i,0)$ rapidly grow. Each function $g_t(i,0)$ is associated with OD flow i and when this function grows, it means that OD flow i is suspected of carrying an abnormal amount of traffic. Contrary to $g_t(i,0)$, only decision function $s_t(159)$ associated to OD flow 159 exceeds the localization threshold. Hence, functions $s_t(i)$ permit to localize the anomalous OD flow among all the OD flows associated to functions $g_t(i,0)$ that have rapidly grown. At time 3660, an alarm is raised and the algorithm selects the anomalous OD flow 159. The decision function $s_t(i)$ needs only 1 measurement to localize the anomalous OD flow. Similar results are obtained in Abilene in Fig. 6c,d, for a volume anomaly in OD flow 87.



(a) Recursive function $g_t(i,0)$ **(b)** Decision function $s_t(i)$

(c) Recursive function $g_t(i,0)$ **(d)** Decision function $s_t(i)$

Fig. 6 Typical realizations of decision functions for **a**,**b** a Tier-2 network and **c**, **d** Abilene. TMs come from the week starting on the 01/05/04 in Abilene, and on the 04/04/05 in the Tier-2 network

6 Reactive Robust Routing

Both proactive and reactive methods, namely MHRR and the anomaly detection/localization algorithm, respectively, are combined into a single approach, introduced in this paper as Reactive Robust Routing (RRR). The reader is referred to the high level description of RRR in Fig. 1. This approach provides an automatic method for robust routing configuration/reconfiguration, based on the monitoring of the network state. RRR exploits the localization ability of the detection/localization algorithm to deploy a new and adapted robust routing configuration after the detection of an anomalous OD flow; at the same time, it detects the end of the anomaly (if there is any) and takes back the usual MHRR configuration.

6.1 Routing Reconfiguration

A simple method that exploits both the SRR approach and the localization ability of the detection/localization algorithm is proposed to compute a new routing configuration to handle a volume anomaly. The idea of this reconfiguration is to minimize the impacts of the detected anomaly on the network performance. As part of MHRR, we shall assume that before the detection of the anomalous traffic, a SRR configuration R_{robust} is used, computed on the basis of some historical routing R_{o} and some measured links traffic $\mathbf{y}_{\text{o}}$.

Let us consider an anomalous increase θ in the traffic volume of OD flow k. The normal-operation traffic demand $\mathbf{d}$ takes the value $\mathbf{d}_{\text{anomaly}} = \mathbf{d} + \boldsymbol{\theta}$, with $\boldsymbol{\theta} = \theta.\boldsymbol{\delta}_k$, where $\boldsymbol{\delta}_k = (\delta_{1,k}, \ldots, \delta_{k,k}, \ldots, \delta_{m,k})^T$, $\delta_{i,k} = 0$ if $i \neq k$ and $\delta_{i,i} = 1$. In RRR, the normal-operation uncertainty set D is expanded in the directions of the routed anomalous OD flow k, obtaining an expanded uncertainty set $D_{\text{anomaly}} = \left\{\mathbf{d}_{\text{anomaly}} \in \mathbb{R}^m, R_{\text{o}}.\mathbf{d}_{\text{anomaly}} \leqslant \mathbf{y}_{\text{o}} + \mathbf{r}_{\text{o}_k}\theta, \mathbf{d}_{\text{anomaly}} \geqslant 0\right\}$. The key issue of this expansion is that the new uncertainty set contains now the anomaly, hence a new robust routing configuration can be computed, with the great advantage of being highly adapted to properly handle this anomalous traffic. Figure 7 shows the expansion of the uncertainty set. In the figure, an anomaly of volume θ occurs in OD flow k, which should span over links i, j, and h according to R_{o}. Note that the expansion of the uncertainty set D is done in the directions provided by the k-th column $\mathbf{r}_{\text{o}_k}$ of R_{o}, and not in the directions provided by the real routing configuration R_{robust}. The reader should bear in mind that the anomalies that we deal with originate outside the network and propagate between origin-destination nodes (e.g. flash crowds, denial of service attacks) and therefore, they do not depend on the particular routing configuration used in the network of analysis; the expansion with respect to R_{o} can be thought as a proactive way of computing an uncertainty set that contains an anomaly in OD flow k.

The new SRR scheme $R_{\text{robust}}^{\text{anomaly}}$ is the solution to (3), using D_{anomaly} as the uncertainty set. To avoid the estimation of the unknown anomalous volume θ, D can be expanded to the limits of links capacity c_i (or some fraction λ chosen by the network operator), in the directions of OD flow k : $D_{\text{anomaly}} = \left\{\mathbf{d}_{\text{anomaly}} \in \mathbb{R}^m, R_{\text{o}}.\mathbf{d}_{\text{anomaly}} \leqslant \boldsymbol{\tau}\right\}$, where $\tau_i = y_{\text{o}}(i)$ if $r_{\text{o}_{i,k}} = 0$, and $\tau_i = \lambda c_i$ if $r_{\text{o}_{i,k}} > 0$. Note that the

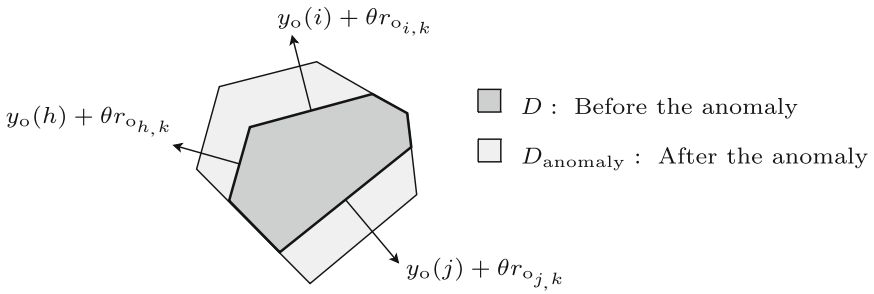


Fig. 7 Robust routing reconfiguration, based on the expansion of the uncertainty set

new routing configuration can be pre-computed off-line for each possible single-OD flow anomaly, obtaining a set of m SRR configurations $R_{\text{robust}}^{\text{anomaly}(k)}, k = 1, \ldots, m$.

Figure 8 presents the evaluation of RRR in the presence of a sudden and abrupt load change. This experiment considers the same situation depicted in Fig. 3b, and compares the routing performance of MHRR and RRR, respectively. Similar to Sect. 3, we assume that the daily uncertainty set is completely known for the case of MHRR (i.e. the abrupt change is known in advance). In RRR, the anomaly is automatically detected and the new routing configuration, computed a-priori on the basis of the expanded uncertainty set, is immediately deployed. The reader can appreciate that the routing performance of RRR is slightly worse than the one obtained by MHRR, less than 2%. Nevertheless, the RRR represents a real scenario, where the anomaly has to be detected in real-time to conduct an accurate routing reconfiguration.

6.2 Back to the MHRR Scenario

In order to regain the MHRR configuration after the end of the detected anomaly, RRR provides a method to detect the return to normal operation. This detection can

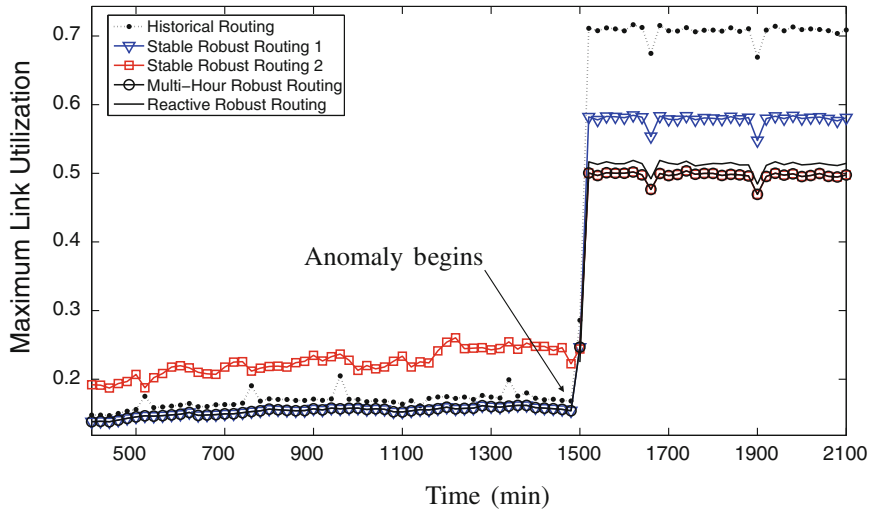


Fig. 8 Reactive Robust Routing—routing reconfiguration after the detection of a large and abrupt traffic change. TMs correspond to traffic demands from the 23th week of the Abilene dataset, starting on the 28/08/04

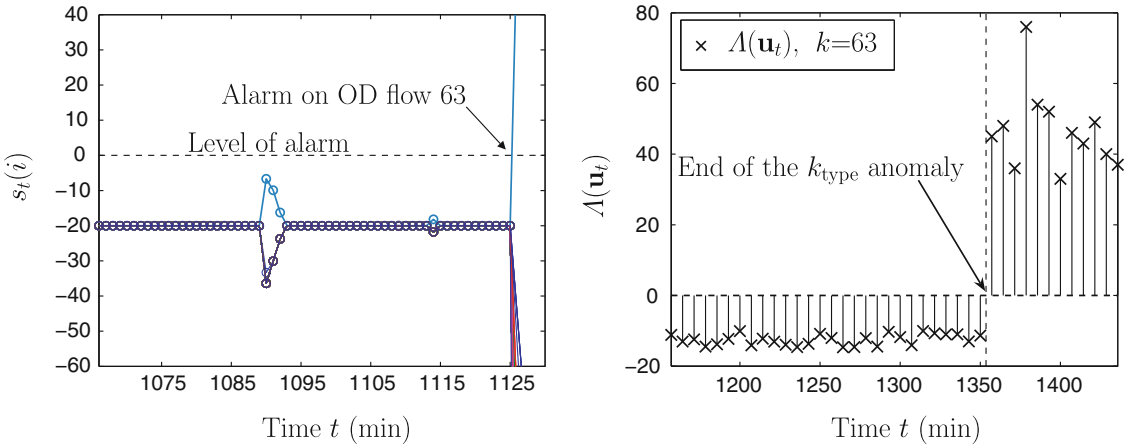
be easily achieved by using a similar hypothesis-testing approach to the one used to detect anomalies: suppose that a k_{type} anomaly has been detected and localized at time t_{anomaly}. For every time $t > t_{\text{anomaly}}$, we focus on the distribution of traffic residuals $f_k(\mathbf{u}_t)$ and look for a change that indicates the end of this anomaly. The reader should remember that in this work we have only considered anomalies in a single OD flow at a single time, and thus the only change that we can expect is back to normal operation.

Two simple hypotheses are considered for this problem, the null hypothesis $\mathcal{H}_0^k = \{\mathbf{u}_t \sim \mathcal{N}(\theta \mathbf{v}_k, I_{r-q})\}$, where the k-th OD flow presents an anomalous additional amount of traffic, against the alternative hypothesis $\mathcal{H}_{\text{alt}} = \{\mathbf{u}_t \sim \mathcal{N}(0, I_{r-q})\}$ where OD flow k is anomaly-free. A Neyman-Pearson test [30] is applied at each time t to decide between $\mathcal{H}_0^k$ and $\mathcal{H}_{\text{alt}}$. The Neyman-Pearson test represents the most powerful test for two simple hypotheses [30]. The statistics of this test is given by:

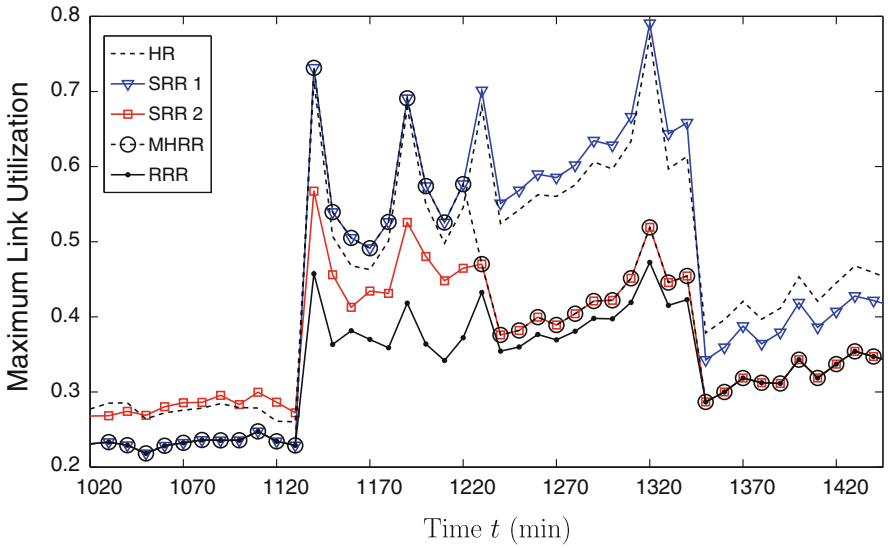
$$\Lambda(\mathbf{u}_t) = \log \frac{f_{\text{alt}}(\mathbf{u}_t)}{f_k(\mathbf{u}_t)} - h \gtrless 0 \tag{24}$$

where the decision threshold h is defined according to the desired false alarm probability. In this case, the function f_{alt} represents the probability density function of residuals under anomaly-free behavior, i.e hypothesis $\mathcal{H}_{\text{alt}}$, while f_k is the probability density function of residuals in the presence of an anomaly in OD flow k, i.e. hypothesis $\mathcal{H}_0^k$. If $\Lambda(\mathbf{u}_t) < 0$, the decision test chooses hypothesis $\mathcal{H}_0^k$. When $\Lambda(\mathbf{u}_t) > 0$, the test decides hypothesis $\mathcal{H}_{\text{alt}}$, pointing out the end of the anomaly.

To conclude this section, Fig. 9 presents an evaluation of the complete RRR approach under the presence of a simulated volume anomaly. An artificial, sudden and large volume change is introduced in OD flow 63 of the Abilene dataset. This artificial traffic is introduced on top of the usual daily traffic between times 1,125 and 1,350. The first step of RRR consists in computing the MHRR configurations, using an expected daily uncertainty set. The optimal division (5) results in $\beta^* = 1{,}230$. The evaluation begins at time 1,020, when MHRR decides to apply R^1_{robust} (SRR 1 in Fig. 9c). The detection/localization algorithm continuously monitors the network state, and at time $t_{\text{anomaly}} = 1{,}125$ detects and localizes an anomalous behavior in OD flow 63 (Fig. 9a). After the detection, and before the new sampling of link loads, the new routing configuration is deployed, which was previously computed using the notions introduced in Sect. 6.1. At time $t = t_{\text{anomaly}} + 1$ the new routing configuration is active, and the anomaly-end detection phase begins. It is important to note that the matrix $H = \Phi^{-\frac{1}{2}} RS$ as well as the anomaly-free traffic *rejector* W must be recomputed after the change of the routing matrix R; in fact, the same re-computation must be conducted every time the routing matrix changes, restarting the detection algorithm to avoid transient effects. The decision statistics $\Lambda(\mathbf{u}_t)$ remains negative for every time $t > t_{\text{anomaly}}$, until time $t' = 1350$, when the positive value of $\Lambda(\mathbf{u}_{t'})$ shows the end of the anomalous behavior in OD flow 63. At this time, RRR compares t' with β^* in order to decide which routing to apply, R^1_{robust} if $t' < \beta^*$ or R^2_{robust} if $t' > \beta^*$. Once the new routing configuration is established, the anomaly detection/localization algorithm starts again to recursively



(a) Anomaly detected in OD flow 63. **(b)** Anomaly-end detected.



(c) Reactive Robust Routing.

Fig. 9 Reactive Robust Routing performance under a simulated volume anomaly. TMs correspond to traffic demands from the 12-th week of the Abilene dataset, starting on the 12/06/04

search for anomalies. The performance improvements of RRR are remarkable, up to 40% w.r.t. MHRR and near 50% w.r.t. the traditional SRR approach.

6.3 Reactive Robust Load Balancing versus Dynamic Load Balancing

Both MHRR and RRR consist in routing reconfiguration. However, even partially modifying the routing configuration of a large-scale network is in the practice a challenging task. It it easy to see that all our routing optimization algorithms can be very easily extended to the case of load balancing, where paths remain always fixed and the only modification is related to the fractions of traffic sent through each of these paths. Such a solution can be easily implemented in the practice, using path-based protocols such as MPLS. We believe that results presented so far will not significantly vary in case of load balancing, mainly because path diversity is rich enough and path modifications were usually rare in the presented evaluations. To

confirm this, we present and evaluate a load balancing extension of RRR, which we shall nickname Reactive Robust Load Balancing (RRLB). The RRLB approach uses a fixed set of paths $P(k)$, $\forall\ k \in N$, and only modifies the routing fractions $r_{p,k}$. Note that such a routing configuration can be directly computed from (3) using $P(k)$ fixed, applying a simpler constrains generation method instead of a combined columns and constraints generation method. The set of paths $P(k)$ used by RRLB is the one obtained from (3) in the computation of the SRR configuration for normal-operation traffic.

Let us evaluate the performance obtained by RRLB when faced with volume anomalies; additionally, we shall compare its execution with the one obtained by a traditional Dynamic Load Balancing algorithm, known as TeXCP [9]. In TeXCP, a convex link congestion function is defined in terms of link capacity and link load. The objective is to minimize the MLU, which is achieved by minimizing the biggest utilization that each OD flow obtains in its paths. This minimization is done recursively by means of a greedy algorithm, in which each single OD flow increases the amount of traffic sent along the path with the smallest utilization, independently of the other $m - 1$ OD flows. Such an adaptive algorithm is periodically executed, based on feedback from the network utilization. This constitutes the most challenging aspect of DLB, because convergence speed might be over-killing, specially under large and abrupt changes in traffic demands.

Figure 10 depicts the MLU obtained by RRLB and TeXCP in the traffic scenario presented in Fig. 8. To be as fair as possible, both algorithms use the same set of paths. In this evaluation, TeXCP adapts routing fractions $r_{p,k}$ every minute, meaning that for each new TM, five updates will be performed (recall that TMs are collected every 5$'$ in Abilene). Results are shown then for every minute. The curve with label 'Actual Minimum' corresponds to the optimal value of MLU, computed for each single TM. Let us begin by TeXCP. A first important observation is that TeXCP has an important overshoot that causes high congestion, with an absolute difference w.r.t. the optimum of about 40%. The convergence after the anomaly is very slow, taking more than 6 h. However, it should be noted that when it eventually

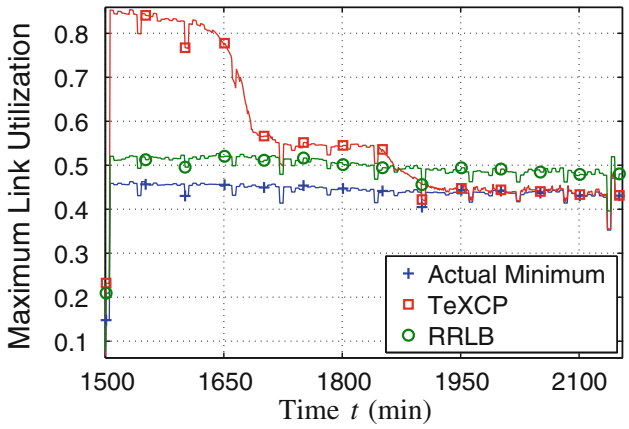


Fig. 10 Reactive Robust Load Balancing (RRLB) versus Dynamic Load Balancing (DLB). Transient behavior of DLB might be over-killing. RRLB avoids transient congestion thanks to the proactive computation of robust load balancing configurations. TMs correspond to traffic demands from the 23th week of the Abilene dataset, starting on the 28/08/04

converges, it obtains an optimal MLU. On the contrary, RRLB does not present a transient congestion behavior, basically because the routing fractions are not continuously modified but precomputed a-priori and immediately applied after the detection of the anomaly. Note that RRLB does not reach optimality, but the performance degradation is only 5%. This simple evaluation shows the great advantage of our approach, which provides a load-balancing technique without highly disruptive transient behaviors. Network operators are reluctant to use dynamic mechanisms mainly because they are afraid of this kind of transient behavior, and as we have seen, these concerns are not without reason. However, RRLB imposes as a plausible solution to the problem.

6.4 Numerical Complexity of RRLB and Implementation Issues

Data collection overhead, numerical complexity, and memory usage are central issues to determine the extra overhead introduced by RRLB. We shall focus on the reactive part of RRLB, namely the anomaly detection/localization algorithm, because the computation of the different SRR configurations used by RRLB is performed off-line, hence it does not limit its applicability.

Data collection is not really an issue in RRLB, for two main reasons: firstly, our algorithms use highly aggregated measurements, namely links traffic measurements, sampled at relatively low rates, e.g., one set of measurements every 5 min in this work, hence the cost of data collection is almost negligible; additionally, SNMP measurements are available in virtually every IP device, and are usually collected in every operational network, thus measurements are readily there to be used, without necessity of additional monitoring technology.

As regards anomaly detection, the method stores two matrices in memory, the "whitening" matrix $\Phi^{-\frac{1}{2}} \in \mathbb{R}^{r\times r}$, and the rejector $W \in \mathbb{R}^{(r-q)\times r}$. Given the recursive execution of the method, m additional variables are kept in memory, corresponding to the m recursive functions $g_t(i,0)$. For anomaly localization purposes, the m anomaly signatures $\mathbf{v}_k \in \mathbb{R}^{(r-q)\times 1}$ are also stored. This represents a total of $\mathcal{O}(r^2)$ variables in memory, where r is the number of links in the network. The computation of $\Phi^{-\frac{1}{2}}$ and W involves matrix multiplications and inversions, and thus the associated cost is $\mathcal{O}(r^3)$. There is an additional cost in the learning phase of the splines-based model, related to the tomogravity estimate used to construct the splines basis S. The cost of the tomogravity method is similar to that of the least-squares method, which implies $\mathcal{O}(m^3)$ operations to estimate an $m \times 1$ TM. All these matrices are computed off-line during the learning phase and do not affect the scalability and on-line applicability of the method. In the on-line detection/localization phase, residuals $\mathbf{u}_t = W\mathbf{z}_t$ are computed and used to update the m recursive functions $g_t(i,0)$. Finally, the m functions $s_t(i)$ used for anomaly localization are computed. These steps involve approximately $\mathcal{O}(r^2)$ operations for anomaly detection and $\mathcal{O}(m^2)$ additional operations for anomaly localization.

To sum up, RRLB keeps in memory a total of $\mathcal{O}(r^2)$ variables and implies $\mathcal{O}(r^2 + m^2)$ operations for on-line anomaly execution. This analysis shows that the overheads of RRLB are quite low, and demonstrates that the method is easily

scalable with the size of the network. All previous SRR approaches do not incur the overheads of RRLB, but they certainly provide higher MLU values or even become unfeasible in the event of volume anomalies (i.e. $u_{\max} > 1$), simply because there is no single routing configuration that can efficiently manage all possible traffic variations; it is clear from our study that some form of dynamism is necessary when faced with highly dynamic traffic.

7 Conclusions

In this paper, we addressed the routing optimization under traffic uncertainty problem. We introduced the Reactive Robust Routing approach, a proactive and reactive solution that not only deals with current dynamic traffic demands in a robust and efficient way, but also detects and localizes volume anomalies from aggregated links traffic measurements, improving network operation. We extended the robust routing technique to manage expected traffic variations by introducing the notion of time-varying uncertainty set, setting up a multi-hour robust routing scheme. This approach achieves better resource utilization than previous stable robust proposals in different scenarios. We introduced an original linear model to parameterize usual traffic behavior from widely available SNMP measurements. Compared to other traffic models, ours is not data-driven, a necessary property to achieve reliable results. In order to manage large and unexpected traffic variations, we presented a statistical algorithm to detect and localize volume anomalies in network traffic, using the proposed model to filter the anomaly-free traffic. This algorithm presents well-established optimality properties, which are extremely important to provide solid results. We proposed a novel approach that exploits both the Robust Routing paradigm and the localization ability of our detection/localization algorithm to optimally handle anomalous traffic variations. Contrary to traditional Dynamic Load Balancing algorithms, the load balancing extension of RRR does not present transient congestion behaviors, a necessary condition to apply dynamic routing in the practice.

Appendix: Elimination of the Anomaly-free Traffic

The anomaly-free traffic $H\boldsymbol{\mu}_t$ is removed by projecting the whitened measurements vector $\mathbf{z}_t = \Phi^{-\frac{1}{2}}\mathbf{y}_t$ onto the left null space of H (i.e. $WH = 0$). Let us define the matrix $W^T = (\mathbf{w}_1, .., \mathbf{w}_{r\text{-}q})$ of size $r \times (r - q)$, composed of eigenvectors $\mathbf{w}_1, .., \mathbf{w}_{r\text{-}q}$ of the projection matrix $P_H^{\perp} = I_r - H(H^TH)^{-1}H^T$, corresponding to eigenvalue 1. The matrix W satisfies the following conditions: $WH = 0, W^TW = P_H^{\perp}$, and $WW^T = I_{r\text{-}q}$. The matrix W can be considered as a linear rejector that eliminates the anomaly-free traffic. In the presence of an anomaly in OD flow j, traffic residuals $\mathbf{u}_t = W\mathbf{z}_t$ can be modeled as $\mathbf{u}_t = \theta W\Phi^{-\frac{1}{2}}\mathbf{r}_j + W\epsilon_t$. By defining the vectors $\mathbf{v}_j = W\Phi^{-\frac{1}{2}}\mathbf{r}_j$, we get the following distribution for traffic residuals in the presence of an anomaly in OD flow j : $\mathbf{u}_t \sim \mathcal{N}(\theta\mathbf{v}_j, I_{r-q})$.

References

1. Cisco Systems: Global IP Traffic Forecast and Methodology, 2006–2011. http://www.hbtf.org/files/cisco_IPforecast.pdf. Accessed 05 Aug 2010
2. Lakhina, A., Crovella, M., Diot, C.: Diagnosing Network-Wide Traffic Anomalies. Proceedings of the SIGCOMM'04 (2004). doi:10.1145/1015467.1015492
3. Ben-Ameur, W., Kerivin, H.: Routing of uncertain traffic demands. Optim. Eng. **6**, 283–313 (2005)
4. Applegate, D., Cohen, E.: Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs. Proceedings of the SIGCOMM'03 (2003). doi:10.1145/863955.863991
5. Wang, H., Xie, H., Qiu, L., Yang, Y., Zhang, Y., Greenberg, A.: COPE: Traffic Engineering in Dynamic Networks. Proceedings of the SIGCOMM'06 (2006). doi:10.1145/1151659.1159926
6. Roughan, M., Thorup, M., Zhang, Y.: Traffic Engineering with Estimated Traffic Matrices. Proceedings of the IMC'03 (2003). doi:10.1145/948205.948237
7. Zhang, C., Liu, Y., Gong, W., Kurose, J., Moll, R., Towsley, D.: On Optimal Routing with Multiple Traffic Matrices. Proceedings of the INFOCOM'05 (2005). doi:10.1109/INFCOM.2005.1497927
8. Casas, P., Vaton, S.: An Adaptive Multi Temporal Approach for Robust Routing. Euro-FGI Workshop on IP QoS and Traffic Control, IST Press, Portugal (2007)
9. Kandula, S., Katabi, D., Davie, B., Charny, A.: Walking the Tightrope: Responsive yet Stable Traffic Engineering. Proceedings of the SIGCOMM'05 (2005). doi:10.1145/1090191.1080122
10. Elwalid, A., Jin, C., Low, S., Widjaja, I.: MATE: MPLS Adaptive Traffic Engineering. Proceedings of the INFOCOM'01 (2001). doi:10.1109/INFCOM.2001.916625
11. Zhang, Y., Ge, Z., Greenberg, A., Roughan, M.: Network Anomography. Proceedings of the IMC'05, 317–330 (2005)
12. Soule, A., Salamatian, K., Taft, N,: Combining Filtering and Statistical Methods for Anomaly Detection. Proceedings of the IMC'05, 331–344 (2005)
13. Ringberg, H., Soule, A., Rexford, J., Diot, C.: Sensitivity of PCA for Traffic Anomaly Detection. Proceedings of the SIGMETRICS'07 (2007). doi:10.1145/1254882.1254895
14. Casas, P., Fillatre, L., Vaton, S., Chonavel, T.: Efficient Methods for Traffic Matrix Modeling and On-line Estimation in Large-Scale IP Networks. Proceedings of the ITC 21 (2009)
15. Casas, P., Fillatre, L., Vaton, S.: Robust and Reactive Traffic Engineering for Dynamic Traffic Demands. Proc. NGI'08 (2008). doi:10.1109/NGI.2008.16
16. Casas, P., Fillatre, L., Vaton, S.: Multi Hour Robust Routing and Fast Load Change Detection for Traffic Engineering. Proceedings of the IEEE ICC'08 (2008) doi:10.1109/ICC.2008.1081
17. Ben-Ameur, W.: Between Fully Dynamic Routing and Robust Stable Routing. Proceedings of the DRCN'07 (2007). doi:10.1109/DRCN.2007.4762277
18. Pioro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Elsevier/Morgan Kaufmann, Amsterdam (2004)
19. Zhang, Y.: Abilene Dataset (2004). http://userweb.cs.utexas.edu/~yzhang/. Accessed 12 Aug 2010
20. Abilene Obs.: http://www.internet2.edu/observatory. Accessed 12 Aug 2010
21. Coates, M., Hero, A., Nowak, R., Yu, B.: Internet tomography. IEEE Signal Process. Mag. **19**(3), 47–65 (2002)
22. Medina, A., Salamatian, K., Bhattacharyya, S., Diot, C.: Traffic Matrix Estimation: Existing Techniques and New Directions. Proceedings of the SIGCOMM'02 (2002). doi:10.1145/964725.633041
23. Zhang, Y., Roughan, M., Duffield, N., Greenberg, A.: Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Load Measurements. Proceedings of the SIGMETRICS'03 (2003). doi:10.1145/781027.781053
24. Nürnberger, G.: Approximation by Spline Functions. Springer, New York (1989)
25. TOTEM: http://totem.run.montefiore.ulg.ac.be/. Accessed 12 Aug 2010
26. Soule, A., Lakhina, A., Taft, N., Papagiannaki, K., Salamatian, K., Nucci, A., Crovella, M., Diot, C.: Traffic Matrices: Balancing Measurements, Inference and Modeling. Proceedings of the SIGMETRICS'05 (2005). doi:10.1145/1071690.1064259
27. Lakhina, A., Crovella, M., Diot, C.: Characterization of Network-Wide Anomalies in Traffic Flows. Proceedings of the IMC'04 (2004). doi:10.1145/1028788.1028813
28. Nikiforov, I.: A lower bound for the detection/isolation delay in a class of sequential tests. IEEE Trans. Inf. Theor. **49**(11), 3037–3047 (2003)

29. Nikiforov, I.: A simple recursive algorithm for diagnosis of abrupt changes in random signals. IEEE Trans. Inf. Theor. **46**(7), 2740–2746 (2000)
30. Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Applications. Prentice Hall, NJ (1993)

Author Biographies

Pedro Casas received an Electrical Engineering degree from the "Universidad de la Republica" (UDELAR) in Montevideo, Uruguay in 2005, and a Ph.D. degree in Computer Sciences from Télécom Bretagne, Brest, France in 2010. He holds a teaching and research Assistant position at UDELAR since 2001, and he is a member of the Electrical Engineering Department at the Engineering Faculty of UDELAR since 2003. He joined the french LAAS-CNRS research laboratory in 2010 as a Postdoctoral Research Fellow. His research interests are related to the statistical characterization and analysis of network traffic, network modeling, anomaly detection, and performance analysis in heterogeneous networks supporting Quality of Service.

Lionel Fillatre received the M.Sc. degree in decision and information engineering and the Ph.D. degree in systems optimization from the University of Technology of Troyes (UTT), France, in 2001 and 2004, respectively. From 2005 to 2007, he worked at Télécom Bretagne, Brest, France, first, as a research engineer in the Computer Science department, then as an Associate Professor in the Signal and Communications department. Since 2007, he is an Associate Professor at the Systems Modeling and Dependability Laboratory, UTT. His current research interests include statistical decision theory, signal and image processing, anomaly detection in traffic flows and information hiding in digital imagery.

Sandrine Vaton obtained the Engineering degree from Télécom Paris in 1994, a M.Sc. degree in applied probabilities from the University of Paris 6 in 1995 and a Ph.D. in signal processing from Télécom Paris in 1998. Since 1999 she is an Associate Professor at Télécom Bretagne in Brest, France. Her main research interests concern statistical network traffic analysis, traffic engineering, Quality of Service, and security in telecommunication networks.

Igor Nikiforov received his M.Sc. degree in automatic control from the Moscow Physical - Technical Institute in 1974, and the Ph.D. in automatic control from the Institute of Control Sciences (USSR Academy of Science), Moscow, in 1981. He joined the University of Technology of Troyes (UTT) in 1995, where he is Professor in the system modeling and dependability laboratory (LM2S), which is a part of the Institute of Charles Delaunay, FRE CNRS 2848. His scientific interests include statistical decision theory, detection/isolation of abrupt changes, fault detection/isolation/reconfiguration, signal, image processing and navigation.

Taming Traffic Dynamics: Analysis and Improvements

Pedro Casas[a,b], Federico Larroca[c,*], Jean-Louis Rougier[c], Sandrine Vaton[a]

[a] *Télécom Bretagne. Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France*
[b] *Facultad de Ingeniería, Universidad de la República. Julio Herrera y Reissig 565, C.P. 11.300, Montevideo, Uruguay*
[c] *Télécom ParisTech. 46 rue Barrault, F-75634 Paris Cedex 13, Paris, France*

Abstract

Internet traffic is highly dynamic and difficult to predict in current network scenarios, which enormously complicates network management and resources optimization. To address this uncertainty in a robust and efficient way, two almost antagonist Traffic Engineering (TE) techniques have been proposed in the last years: Robust Routing and Dynamic Load-Balancing. Robust Routing (RR) copes with traffic uncertainty in an off-line preemptive fashion, computing a single static routing configuration that is optimized for traffic variations within some predefined uncertainty set. On the other hand, Dynamic Load-Balancing (DLB) balances traffic among multiple paths in an on-line reactive fashion, adapting to traffic variations in order to optimize a certain congestion function. In this article we present the first comparative study between these two alternative methods. We are particularly interested in the performance loss of RR with respect to DLB, and on the response of DLB when faced with abrupt changes. This study brings insight into several RR and DLB algorithms, evaluating their virtues and shortcomings, which allows us to introduce new mechanisms that improve previous proposals.

Keywords: Traffic Uncertainty, Traffic Management, Robust Optimization, Robust Routing, Dynamic Load-Balancing

1. Introduction

As network services and Internet applications evolve, network traffic is becoming increasingly complex and dynamic. The convergence of data, telephony and television services on an all-IP network directly translates into a much higher variability and complexity of the traffic injected into the network. To make matters worse, the presence of unexpected events such as network equipment failures, large-volume network attacks, flash crowd occurrences and even external routing modifications induces large uncertainty in traffic patterns. Moreover, current evolution and deployment-rate of broadband access technologies (e.g. Fiber To The Home) only aggravates this uncertainty.

But these are not the only problems network operators are confronted with. The ever-increasing access rates available for end-users we just mentioned is such that the assumption of infinitely provisioned core links could soon become obsolete. In fact, recent Internet traffic studies from major network technology vendors like Cisco Systems forecast the advent of the Exabyte era [1, 2], a massive increase in network traffic driven by high-definition video. In this context, simply upgrading link capacities may no longer be an economically viable solution. Moreover, even if overdimensioning would be possible, its environmental impact is not negligible. For instance, the Information and Communication Technology sector alone is responsible for around 2% of the man-made CO_2, a similar figure to that of the airline industry, but with higher increasing perspectives [3]. An efficient and responsible usage of the resources is then essential[1].

In the light of this traffic scenario, we study the problem of intradomain Traffic Engineering (TE) under traffic uncertainty. This uncertainty is assumed to be an exogenous traffic modification, meaning that traffic variations are not produced within the domain for which routing is optimized but are due to external and difficult to predict events. More in particular, we are interested in two almost antagonist approaches that have emerged in the recent years to cope with both the increasing traffic dynamism and the need for cost-effective solutions: Robust Routing (RR) [6, 7, 8] and Dynamic Load-Balancing (DLB) [9, 10, 11].

In RR, traffic uncertainty is taken into account directly within the routing optimization, computing a single routing configuration for all traffic demands within some *uncertainty set* where traffic is assumed to vary. This uncertainty set can be defined in different ways, depending on

*Corresponding author. Telephone: +33 (0)1 45 81 75 52 - Fax: +33 (0)1 45 89 79 06

Email addresses: pedro.casas@telecom-bretagne.eu (Pedro Casas), federico.larroca@telecom-paristech.fr (Federico Larroca), rougier@telecom-paristech.fr (Jean-Louis Rougier), sandrine.vaton@telecom-bretagne.eu (Sandrine Vaton)

[1] To learn more about this emerging discipline, the interested reader should consult works related to so-called "green networking" [4].

the available information: largest values of links load previously seen, a set of previously observed traffic demands (previous day, same day of the previous week), etc. The criterion to search for this unique routing configuration is generally to minimize the maximum link utilization (i.e. the utilization of the most loaded link in the network) for all traffic demands of the corresponding uncertainty set. While this routing configuration is not optimal for any single traffic demand within the set, it minimizes the worst case performance over the whole set.

DLB copes with traffic uncertainty and variability by splitting traffic among multiple paths in real-time. In this dynamic scheme, each origin-destination (OD) pair of nodes within the network is connected by several a priori configured paths, and the problem is simply how to distribute traffic among these paths in order to optimize a certain function. DLB is generally defined in terms of a link-congestion function, where the portions of traffic are adjusted in order to minimize the total network congestion. Ideally, the traffic distribution is set so that at every instant the objective function is optimized.

Those who promote DLB highlight among others the fact that it is the most resource-efficient possible scheme, and that given the configured paths it supports every possible traffic demand, all of this in an automated and decentralized fashion. In practice, the "always-optimized" characteristic we mentioned above is achieved by means of a distributed algorithm periodically executed by every ingress router based on feedback from the network. It is precisely this last characteristic that constitutes the most challenging aspect of DLB. In fact, the deployment of DLB has been, to say the least, limited. Two particular problems arise in DLB: convergence to the optimum is not always guaranteed, and convergence speed might be over-killing under large and abrupt changes in traffic demands. Network operators are reluctant to use dynamic mechanisms mainly because they are afraid of a possible oscillatory behavior of the algorithm used by each OD pair to adjust load-balancing. As the early experiences in ArpaNet has proved [12], these concerns are not without reason. (In particular, before July 1987, the links' metric was defined as the packet delay averaged over a 10 s period. Although this adaptive routing scheme worked correctly under light or moderate loads, it generated oscillations under relatively heavy loads. This resulted in substituting the links' metric by a fixed value as we use it today, sacrificing optimality for stability.) Indeed, for these adaptive and distributed algorithms, a trade-off between adaptability (convergence speed) and stability must be found, which may be particularly difficult in situations where abrupt traffic changes occur.

Those who advocate the use of RR claim that there is actually no need to implement supposedly complicated and possibly oscillatory dynamic routing mechanisms, and that the incurred performance loss for using a single routing configuration is negligible when compared with the increase in complexity. RR provides a stable routing configuration for all the traffic demands within the uncertainty set, avoiding possible oscillations and convergence issues. However, RR presents some conception problems and serious shortcomings in its current state which we highlight and try to ease in this work. The first drawback of current RR is related to the objective function it intends to minimize. Optimization under uncertainty is generally more complex than classical optimization, which forces the use of simpler optimization criteria such as maximum link utilization (MLU). The MLU is not the most suitable network-wide optimization criterion; setting the focus too strictly on MLU often leads to worse distribution of traffic, adversely affecting the mean network load and thus the total network end-to-end delay, an important QoS indicator. It is easy to see that the minimization of the MLU in a network topology with heterogeneous link capacities may lead to poor results as regards global network performance. The second drawback of RR we identify is its inherent dependence on the definition of the uncertainty set of traffic demands: the uncertainty set has to be sufficiently "large" to allow traffic flexibility and to provide performance guarantees, but should not be excessively "large" to avoid wasting network resources. Thus, considering a unique RR configuration to address both traffic in normal operation and unexpected traffic variations is an inefficient strategy, as a single routing configuration cannot be suitable for both situations.

1.1. Contributions of this article

This article presents a fair and comprehensive comparative analysis between RR and DLB mechanisms. The analysis is comprehensive as it evaluates the performance of both mechanisms based on different performance indicators and considering normal operation as well as unpredicted traffic events. We believe our comparison is fair because it considers the particular characteristics of each mechanism under the same network and traffic conditions. To date and to the best of our knowledge this is the first work that conducts such a comparative evaluation, necessary indeed not only from a research point of view but also for network operators who seek cost-effective and robust solutions to face future network scenarios. Based on this comparative analysis we develop and evaluate new variants of RR and DLB mechanisms, improving some of the shortcomings found in both static and dynamic approaches.

Regarding the RR approach, we will introduce some modifications that strive to alleviate the two problems identified in current proposals. We will first study which is the best objective function to minimize, and propose the mean link utilization instead of the MLU. The mean link utilization provides a better image of network-wide performance, as it does not depend on the particular load or capacity of each single link in the network but on the average value. However, a direct minimization of the mean link utilization does not assure a bounded MLU, which is not practical from an operational point of view. Thus, we minimize the mean link utilization while bounding the MLU

by a certain utilization threshold a priori defined. This adds a new, and maybe difficult to set, constraint to the problem, namely how to define this utilization threshold. We further improve our proposal by providing a multiple objective optimization criterion, where both the MLU and the mean link utilization are minimized simultaneously. We evaluate the improvements of our proposals from a QoS perspective, using the mean path end-to-end queuing delay as a measure of global performance.

The second problem we address in RR is the trade-off between routing performance and routing reliability. In [13] we have recently proposed a solution to manage this trade-off, known as Reactive Robust Routing (RRR). Basically, RRR consists of constructing a RR configuration for expected traffic in nominal operation, adapting this nominal routing configuration after the detection and localization of a large and long-lived traffic modification. RRR provides good performance for both nominal operation and unexpected traffic, but it is difficult to deploy in a real implementation, because of the routing reconfiguration step. Reconfiguring the routing of an entire Autonomous System is a nontrivial task. In this article we modify the RRR approach, using a preemptive Load Balancing algorithm to balance traffic among pre-established paths after the localization of a large volume traffic modification (preemptive in the sense of preventing a situation from occurring).

In what respects DLB, we evaluate the use of so-called *no-regret* algorithms as the distributed optimization algorithm used by ingress routers to adapt load-balancing. The authors of a recent paper [14] proved that if all OD pairs use algorithms of this kind, convergence to the optimum is guaranteed. Special attention will be paid on the behavior of the algorithm when faced with abrupt and unexpected changes in the traffic demands. We shall introduce simple, and yet effective, modifications to the algorithm to assure a fast convergence to the new optimum in this case.

As we shall see in the following subsection, several previously proposed DLB algorithms strive to minimize the MLU by means of a greedy algorithm in the paths utilization (i.e. each ingress router increases the amount of traffic sent along the path with the smallest utilization). As proved in a recent paper [15], convergence to the optimum for such algorithms may not be guaranteed, in the sense that they may converge to a situation in which the MLU is not minimized. In fact, we shall present an example in which the difference with the optimum of the MLU is non-negligible. In this work we will present an alternative path cost function, so that greedy algorithms that use it do converge to the optimum equilibrium.

1.2. Related Work

There is a large literature on routing optimization with uncertain traffic demands. Thus, here we shall only mention a few papers, and do not expect our list to be exhaustive.

Traditional algorithms rely on a single or a small group of expected traffic demands to compute optimal and reliable routing configurations. An extreme case is presented in [16], where routing is optimized for a single estimated traffic demand and is then applied for daily routing. Traffic uncertainty is characterized by multiple traffic demands in [17] (set of traffic demands from previous day, same day of previous week, etc.), where different mechanisms to find optimal routes for the set are presented. As discussed for instance in [18], this perspective is no longer suitable for current and future dynamic scenarios. These approaches require a "leap of faith" to perform well, mainly because they assume that traffic patterns do not change that much over time. However, even a relatively small difference between the "real" traffic demand and the one used for the routing optimization may lead to an important performance degradation. Such a difference may arise in the event of unexpected traffic variations (which are more common nowadays), or even be due to an error in the traffic estimation.

A different approach has emerged in the recent years to cope with the increasing traffic dynamism and the need for cost-effective solutions, Dynamic Load-Balancing (DLB) [9, 10, 11, 19]. In DLB, traffic is split among a priori established paths in order to avoid network congestion. The two most well-known proposals in this area are MATE and TeXCP. In MATE [9], a convex link congestion function is defined, which depends on the link capacity and the link load. The objective is to minimize the total network congestion, for which a simple gradient descent method is proposed. In [19], we propose to use a link congestion function based on measurements of the queueing size, which results in better global performance from a QoS perspective. TeXCP [10] proposes a somewhat simpler objective: in order to minimize the MLU, they minimize the biggest utilization each traffic demand obtains in its paths. Another DLB scheme which has the same objective but a relatively different mechanism is REPLEX [11].

The last category of algorithms consists of Robust Routing techniques [6, 7, 8, 20, 21]. The objective in RR is to find a unique static routing configuration that fulfills a certain criterion for a broad set of traffic demands, generally the one that minimizes the maximum link utilization over the whole set of demands. In [6], authors capture traffic variations by introducing a polyhedral set of demands, which allows for easier and faster linear optimization. This robust technique is applied in [20] to compute a robust MPLS routing configuration without depending on traffic demand estimation, and corresponding methods for robust OSPF optimization are discussed. Oblivious Routing [7] also defines linear algorithms to optimize worst-case MLU for different sizes of traffic uncertainty sets. The author of [21] analyzes the use of robust routing through a combination of traffic estimation techniques and its corresponding estimation error bounds, in order to shrink the set of traffic demands. In [8] authors introduce COPE, a RR mechanism that optimizes routing for

predicted demands and bounds worst-case MLU to ensure acceptable efficiency under unexpected traffic events. The idea behind COPE is similar to ours, in the sense that it strives to alleviate performance degradation due to unforeseen traffic modifications. Nevertheless, it proposes a single routing configuration to handle expected as well as large and abrupt traffic variations, which is clearly not the best solution.

The same paper [8] presents, to the best of our knowledge, the only previous comparative study between RR and DLB. The authors of the paper compare the performance of COPE with a dynamic approach which they claim models the behavior of mechanisms such as MATE and TeXCP. Given a time series traffic demands, this dynamic approach consists of computing an optimal routing for each traffic demand i and evaluate its performance with the following traffic demand $i+1$. There are two important shortcomings of this DLB simulation. Firstly, adaptation in DLB is iterative and never instantaneous. Secondly, in all DLB mechanisms paths are set a priori and remain unchanged during operation. This is not the case in their dynamic approach, where each new routing optimization may change not only traffic portions but paths themselves. For these reasons, we believe that the comparison provided in [8] is biased against dynamic schemes.

The remainder of this article is organized as follows. In Sec. 2 we introduce the network model and notation, while Sec. 3 and 4 introduce a preliminary version of the RR and DLB mechanisms. Some first results are discussed in Sec. 5. Section 6 presents new variants to the former mechanisms which alleviate the shortcomings detected. The evaluation of the complete set of algorithms under different traffic scenarios is conducted in Sec. 7. We finally draw conclusions of this comparative analysis in Sec. 8.

2. Network Model and Performance Indicators

Let us begin by introducing the notation used in this article. The network topology is defined by n nodes and a set $L = \{l_1, \ldots, l_q\}$ of q links, each with a corresponding capacity c_i, $i = 1, \ldots, q$. The Traffic Matrix (TM) $X = \{x_{i,j}\}$ denotes the traffic demand (expressed in, for example, Mbps) between every origin node i and every destination node j ($i \neq j$) of the network; we shall note each of these origin-destination pairs as OD pairs, and each origin-destination traffic demand $x_{i,j}$ as OD flows. In practice, each traffic demand is measured every T minutes (usually 5' or 10' [5]), and its value simply corresponds to the cumulative number of bytes observed between two consecutive measurements, divided by the polling time T. Let $X = \{x_k\}$ be the vector representation of the TM, where we have reordered OD flows by index $k = 1, \ldots, m$ ($m = n.(n-1)$). Let $N = \{\mathrm{OD}_1, \ldots, \mathrm{OD}_m\}$ be the set of m OD pairs. We consider a multi-path network topology, where each OD flow x_k can be arbitrarily split among a set of p_k origin-destinations paths P_k. In this sense, we shall call r_p^k the portion of traffic flow x_k sent along path $p \in P_k$, where $0 \leqslant r_p^k \leqslant 1$ and $\sum_{p \in P_k} r_p^k = 1$.

Let λ_l^p be an indicator variable that takes value 1 if path p traverses link l and 0 otherwise, and $Y = \{\rho_1, \ldots, \rho_q\}$ a vector representation of links traffic load. Then X and Y are related through the routing matrix R, a $q \times m$ matrix $R = \{r_l^k\}$ where $r_l^k = \sum_{p \in P_k} \lambda_l^p \,.\, r_p^k$. The variable r_l^k indicates the fraction of OD flow x_k routed along link l; this results in the following relation:

$$Y = R \,.\, X \tag{1}$$

Given X, the multi-path routing optimization problem consists in choosing the set of paths P_k for each OD pair k and computing the routing matrix R, in order to optimize a certain objective function $g(X, R)$. A simplified version of this problem is the load-balancing optimization problem which, *given* a set of paths, calculates R. In this work we shall consider different performance indicators, which result in different objective functions.

A very important link-level performance indicator is the link utilization $u_l = \rho_l / c_l$; a value of u_l close to one indicates that the link is operating near its capacity. Network operators usually prefer to keep links utilization relatively low in order to support sudden traffic increases and link/node failures. A network-wide performance indicator is the maximum link utilization $u_{\max}$:

$$u_{\max}(X, R) = \max_{l \in L} \{u_l\} \tag{2}$$

The maximum link utilization constitutes by far the most popular TE objective function. However, its minimization presents a clear drawback: setting the focus too strictly on the most utilized link often leads to a worse distributions of traffic, adversely affecting the overall performance in the network. In this sense we will consider the mean link utilization u_{mean} as another possible objective function:

$$u_{\mathrm{mean}}(X, R) = \frac{1}{q} \sum_{l \in L} u_l \tag{3}$$

Minimizing u_{mean} may provide better network-wide performance, as long as the maximum link utilization remains bounded; we will further discuss this issue in Sec. 6.

The last performance indicator we shall consider in this work is the queuing delay (i.e. the time that spans between the moment a byte enters the router and leaves it). This choice is justified by two aspects. Firstly, its algebra is relatively simple, in the sense that the total delay of a path is the addition of the delay at each link. Secondly, it is a very versatile indicator. A big queueing delay means more delay and jitter for streaming traffic. Moreover, a link with an important queueing delay is traversed by several bottlenecked flows, meaning that elastic traffic may obtain better throughput in other, less loaded, links. The mean queueing delay may be regarded then as a numerical value of the congestion on the link.

Assume then that the queuing delay on link l is given by the function $d_l(\rho_l)$. Given this function we can compute the queuing delay of path p as $d_p = \sum_{l \in p} d_l(\rho_l)$. As a measure of the network-wide performance, we consider the expected end-to-end (e2e) queuing delay d_{mean} defined as:

$$d_{\text{mean}}(X, R) = \sum_{k \in N} \sum_{p \in P_k} \left(r_p^k \,.\, x_k\right) d_p = \sum_{l \in L} \rho_l \,.\, d_l(\rho_l) \quad (4)$$

That is to say, d_{mean} is a weighted mean e2e queuing delay, where the weight for each path is how much traffic is sent along it ($r_p^k \,.\, x_k$), or in terms of links, the weight for each link is how much traffic is traversing it (ρ_l). We prefer a weighted mean queuing delay to a simple total delay because it reflects more precisely performance as perceived by traffic. Two situations where the total delay is the same, but in one of them most of the traffic is traversing heavily delayed links should not be considered as equivalent. Note that, by Little's law, the value $f_l(\rho_l) = \rho_l \,.\, d_l(\rho_l)$ is proportional to the volume of data in the queue of link l. We will then use this last value as the addend in the last sum in (4), since it is easier to measure than the queuing delay. Finally, note that, differently to $u_{\max}$, a large mean e2e queuing delay translates into bad performance for the majority of the traffic and not only for the traffic that traverses a particularly loaded link.

Based on these definitions we will introduce the different optimization algorithms that strive to minimize some of these performance indicators, considering either the RR or DLB approach.

3. Stable Robust Routing

Finding a multi-path routing configuration minimizing $u_{\max}$ is an instance of the classical multi-commodity flow problem which can be formulated as a linear program [22]. For a single known traffic matrix X, the problem can be easily solved by linear programming techniques [23]. However, as we have previously discussed, traffic demands are uncertain and difficult to predict, and all we can expect is to find them within some bounded uncertainty set.

In a robust perspective of the multi-path routing optimization problem, demand uncertainty is taken into account within the routing optimization, computing a single routing configuration for all demands within some uncertainty set. In this work we consider a polyhedral uncertainty set $\mathbb{X}$, more precisely a *polytope* as in [6], based on the intersection of several half-spaces that result from linear constraints imposed to traffic demand.

As an example, let us define an uncertainty set $\mathbb{X}$ based on a given routing matrix R_{o} and the peak-hour links traffic load Y^{peak} obtained with this routing matrix:

$$\mathbb{X} \;=\; \left\{X \in \mathbb{R}^m, R_{\text{o}}.X \leqslant Y^{\text{peak}},\; X \geqslant 0\right\}$$

Observe that this definition of the uncertainty set has a major advantage: routing optimization can be performed from easily available links traffic load Y without even knowing the actual value of the traffic demand X. Figure 1 depicts the obtained uncertainty set, based on the convex intersection of q half-spaces of the form $\mathbf{r}_{\text{o}}^i \cdot X \leqslant \rho_i^{\text{peak}}$, $\forall i \in L$, where $\mathbf{r}_{\text{o}}^i$ stands for the i-th row of the routing matrix R_{o}.

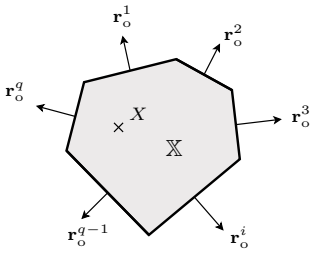


Figure 1: The uncertainty set $\mathbb{X}$ as a polytope.

minimize	$u_{\max}$		
subject to:			
$\sum_{k \in N} \sum_{p \in P_k} \lambda_l^p \,.\, r_p^k \,.\, x(k)$	$\leqslant$	$u_{\max}.c_l$	$\forall\, l \in L,\ \forall\, X \in \mathbb{X}$
$\sum_{p \in P(k)} r_p^k$	$=$	1	$\forall\, k \in N$
r_p^k	$\geqslant$	0	$\forall\, p \in P_k,\ \forall\, k \in N$
$u_{\max}$	$\leqslant$	1	

Table 1: The Robust Routing Optimization Problem (RROP)

The traditional *Robust Routing Optimization Problem* (RROP) defined in Table 1 consists of minimizing the maximum link utilization $u_{\max}$, considering all demands within $\mathbb{X}$. The solution to the problem is twofold: on the one hand, a routing configuration R_{robust}, and on the other hand, a worst-case performance threshold $u_{\max}^{\text{robust}}$:

$$R_{\text{robust}} \;=\; \operatorname*{argmin}_{R} \max_{X \in \mathbb{X}} u_{\max}(X, R)$$

$$u_{\max}^{\text{robust}} \;=\; \max_{X \in \mathbb{X}} u_{\max}(X, R_{\text{robust}})$$

Given a suitable definition of the uncertainty set, the obtained robust routing configuration R_{robust} is applied during long periods of time; in this sense, we refer to Robust Routing as *Stable Robust Routing* (SRR). The authors of [6] have shown that the RROP can be efficiently solved by linear programming techniques, applying a combined columns and constraints generation method. This method iteratively solves the problem, progressively adding new constraints and new columns to the problem.

The new constraints are the extreme points of the uncertainty set $\mathbb{X}$, and the new columns represent new paths added to reduce the objective function value. Only extreme points of $\mathbb{X}$ are added as new constraints, as it is easy to see that every traffic demand $X \in \mathbb{X}$ can be expressed as a linear combination of these extreme demands. Regarding new added paths, the algorithm in [6] may not be the best choice from a practical point of view since the number of paths for each OD pair is not a priori restricted

and the characteristics of added paths are not controlled. For example, it would be interesting to have disjoint paths to route traffic from each single OD pair, improving resilience. For this reason we modify the algorithm to select new paths, both limiting the maximum number of paths in P_k and taking as new candidates the shortest paths with respect to link weights w_l^i:

$$w_l^i = \frac{1}{\epsilon + \left(1 - r_l^{k\,i}\right)} \tag{5}$$

where $r_l^{k\,i}$ corresponds to the fraction of traffic flow x_k that traverses link l after iteration i and ϵ is a small constant that avoids numerical problems. If OD pair k uses a single path p at iteration i, $r_l^{k\,i} = 1$ for every link $l \in p$, and so this path is removed from the graph where new shortest paths are computed ($w_l \to \infty, \forall l \in p$). While this may result in a sub-optimal performance, it allows a real and practical implementation. In case there are no disjoint paths for OD pair k, we use the column constraint generation method used in [6] to add new paths for OD pair k.

4. Dynamic Load Balancing

4.1. Routing Games and Wardrop Equilibrium

As mentioned before, the objective in DLB is to minimize a certain objective function $g(X, R)$ in a distributed fashion (i.e. without relying on any centralized entity). Algorithms that achieve this are typically greedy, which present the desirable property of requiring minimum coordination among border routers. In this kind of mechanisms, a path cost function ϕ_p is defined, and each OD pair greedily minimizes the cost it obtains from each of its paths. This context constitutes an ideal case study for game theory, and is known as *Routing Game* in its terminology [24, 25].

Since each OD pair may arbitrarily balance traffic among its paths, we will assume that OD pairs are constituted of infinitely many agents. These agents control an infinitesimal amount of traffic, and decide along which path to send their traffic. In this context r_p^k represents then the fraction of agents of OD pair k that have p as their choice. If each of these agents acts selfishly, then the system will be at equilibrium when no agent may decrease its cost by unilaterally changing its path decision. This situation constitutes what is known as a *Wardrop Equilibrium* (WE) [26], which is formally defined as follows:

Definition 1. The paths vector $\{r_p^k\}_{k\in N, p\in P_k}$ is a Wardrop Equilibrium if for each OD pair $k \in N$ and for each couple of paths $p, q \in P_k$ with $r_p^k > 0$ it holds that $\phi_p \leq \phi_q$.

Intuitively speaking, a WE is a situation where each OD pair uses only those paths with minimum cost (for the given OD pair). Anyway, the path cost ϕ_p is in turn defined in terms of a certain nonnegative, nondecreasing and continuous link cost function $\phi_l(\rho_l)$. There are roughly two kinds of games depending on the definition of ϕ_p. A *Congestion Routing Game* defines the path cost as $\phi_p = \sum_{l\in p} \phi_l(\rho_l)$. On the other hand, a *Bottleneck Routing Game* defines $\phi_p = \max_{l\in p} \phi_l(\rho_l)$.

Much effort has been put into characterizing the resulting equilibrium of these games. In this sense, a certain social cost function is defined, which measures the dissatisfaction of the OD pairs as a whole (i.e. an optimum paths vector is one that minimizes this function), and the objective is to quantify the difference between the optimum and the resulting WE. In the case of a congestion game, the typical social cost function is the same as in (4) (i.e. $\sum_{l\in L} \rho_l d_l(\rho_l) := \sum_{l\in L} f_l(\rho_l)$), whereas for a bottleneck game the social cost is usually the maximum $\phi_l(\rho_l)$ over all links (i.e. $\max_{l\in L} \phi_l(\rho_l)$).

It may be proved that the WE of a congestion game coincides with the unique minimum of the so-called potential function $\Phi(R) = \sum_{l\in L} \int_0^{\rho_l} \phi_l(x)dx$ [24]. This means that if $f_l(\rho_l)$ is continuous differentiable, non-decreasing and convex, the WE of a congestion game with $\phi_l(\rho_l) = f_l'(\rho_l)$ is socially optimum. In this sense, to minimize d_{mean} through DLB, we will play a Congestion Routing Game with a link cost equal to the derivative of the link mean queue size. In the sequel we shall note this game as MinDG (Minimum Delay Game).

On the other hand, characterization of the WE of a bottleneck game is somewhat more complicated. In fact, it is relatively easy to see that in this case the WE is not even unique. Moreover, and rather unfortunately, it has been proved in [15] that even if there always exist at least one WE that is socially optimum, nothing may be guaranteed about the rest (if any). However, the same paper proved that every WE that fulfills the so-called *efficiency condition* is optimum, where this condition is defined as follows:

Definition 2. Let $B(p)$ denote the number of network bottlenecks over p; that is to say $B(p) = \left|\left\{l \in p : \phi_l(\rho_l) = \max_{m\in L, \rho_m>0} \{\phi_m(\rho_m)\}\right\}\right|$. Then, a WE is said to satisfy the efficiency condition if all OD pairs route their traffic along paths with a minimum number of network bottlenecks; i.e. for all $k \in N$ and $p, q \in P_k$ with $r_p^k > 0$ it holds that $B(p) \leq B(q)$.

This result, which is relatively new, was not applied in the design of neither TeXCP or REPLEX, both of which strive to minimize the maximum link utilization by means of a greedy algorithm in the path utilization (i.e. a bottleneck game with $\phi_p = \max_{l\in p} u_l$). It could then be the case that these algorithms converge to a sub-optimal WE. Possible consequences on the obtained performance of ignoring this result will be further discussed later in the article. In any case, we shall note this game as MinUG (Minimum Utilization Game).

4.2. No-Regret Algorithms

We will now briefly discuss how, given the path cost function ϕ_p, the WE may be achieved for both routing games. In a recent article [14], the authors proved that if all OD pairs use *no-regret* algorithms, the global behavior will approach the WE. To be more precise, for a given TM X, and for *most* time steps, the instantaneous paths vector $\{r_p^k\}_{k\in N, p\in P_k}$ is very close to the WE, and this difference vanishes with time.

This result is very general, in the sense that it does not specify any algorithm in particular. Its only requirement is the use of no-regret algorithms by all OD pairs (for an overview of some of them see [27]). In particular, we will consider the *Weighted Majority Algorithm* (WMA)[28], which originated in the context of online learning (more precisely from the *online prediction using expert advice* problem), and whose pseudo-code for OD pair k is described in Algorithm 1.

Algorithm 1 Weighted Majority Algorithm (WMA)

```
for t = 1, ..., ∞ do
    Obtain path costs φ_p ∀p ∈ P_k
    for every path p ∈ P_k do
        if φ_p > min_{q∈P_k} φ_q then
            r_p^k ← β × r_p^k
        end if
    end for
    Normalize the r_p^k
end for
```

At each iteration t, those paths whose cost is bigger than the minimum are punished by multiplying their respective r_p^k by a certain constant $\beta < 1$ (throughout our simulations we have used $\beta = 0.95$, a value that we empirically verified obtains very good results). Actually, and in order to avoid unnecessary changes in the traffic distribution, we shall only update r_p^k when the corresponding path cost is bigger than the minimum cost plus a certain margin (in the case of MinUG we fixed the margin at 0.005, and for MinDG we used 5% of the minimum).

5. A Preliminary Comparison

In this section we shall present some first simulations that will help us to gain insight into the mechanisms and highlight some of their respective shortcomings. Before, we will discuss how we performed these and the rest of the simulations.

As the reference network we used Abilene, a high-speed Internet2 backbone network. Abilene consists of 12 router-level nodes connected by 30 optical links (we only consider intra-domain links). The used router-level network topology and traffic demands are available at [29]. Traffic data consists of 6 months of traffic matrices collected every 5 minutes via Netflow from the Abilene Observatory [30]. As measured traffic demands do not significantly load the network, we re-scaled them by multiplying all their entries by a constant. The dataset in [29] also provides the static routing configuration R_o deployed in Abilene during the 6-month long TMs measurement campaign.

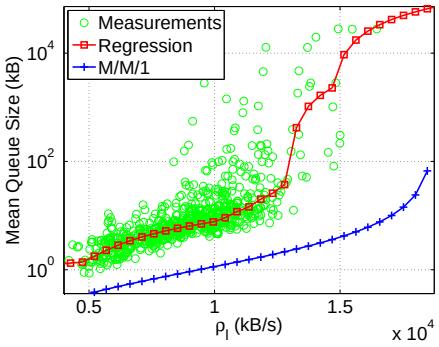


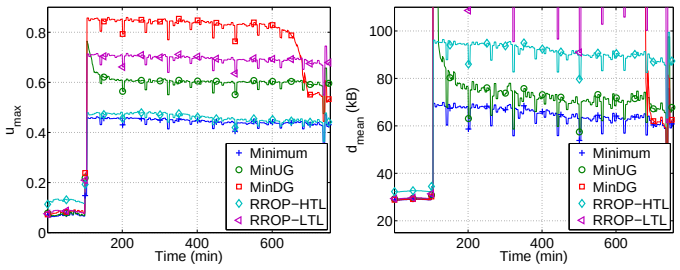
Figure 2: Mean queue size: measurements and approximations

In the case of MinDG, $f_l(\rho_l)$ is typically chosen based on a simplistic model (e.g. the M/M/1 model which yields $f_l(\rho_l) = \rho_l/(c_l - \rho_l)$) [9]. In order to avoid such arbitrary and unprecise choice, in [19] we proposed instead to *learn* this function (and its derivative) from measurements. Figure 2 depicts the real mean queue size of an operational network link at Tokyo obtained from [31], together with the *M/M/1* estimation $f_l^{M/M/1}(\rho_l)$ and the non-parametric regression $\hat{f}_l(\rho_l)$. It is clear that $f_l^{M/M/1}(\rho_l)$ consistently underestimates the real queue size value, while $\hat{f}_l(\rho_l)$ provides quite accurate results.

To be as fair as possible, all mechanisms use the same set of paths, namely those calculated by SRR as discussed in Sec. 3. The TMs are fed to the mechanisms in consecutive temporal order. Both DLB mechanisms (MinUG and MinDG) are initiated at arbitrary values of r_p^k, which will be updated as new link load measurements arrive. We have assumed that each OD pair receives these measurements every minute, meaning that for each new TM five updates of their corresponding r_p^k values will be performed (recall that TMs are collected every 5 min). Results are shown then for every minute. As a reference, we also computed the optimum values $u_{\max}^{\text{opt}}$ and $d_{\text{mean}}^{\text{opt}}$ for every TM X of the dataset.

In this example we consider a traffic scenario that presents an abrupt and large volume increase due to an external routing modification. This corresponds to the TMs with indexes between 1050 and 1200 from dataset X23 in [29]. The evaluation starts with a normal low traffic load situation, but after the 100th minute one of the OD flows abruptly increases its traffic volume, loading the links it traverses until the end of the evaluation.

Regarding SRR, in what follows we shall use the term RROP as a reference to SRR, recalling that the robust routing optimization problem is the one described in Table 1. Based on the static routing matrix of Abilene R_o we define two different polytopes, the former adapted to the



(a) Maximum link utilization (b) Mean e2e queuing delay

Figure 3: Maximum link utilization and mean end-to-end queuing delay. Traffic demand volume abruptly increases after the 100th minute.

Low Traffic Load period (LTL period, before the 100th minute) and the latter adapted to the High Traffic Load period (HTL period, after the 100th minute):

$$\mathbb{X}^{\text{LTL}} = \{X \in \mathbb{R}^m, R_o.X \leqslant Y^{\text{LTL}},\ X \geqslant 0\}$$
$$\mathbb{X}^{\text{HTL}} = \{X \in \mathbb{R}^m, R_o.X \leqslant Y^{\text{HTL}},\ X \geqslant 0\}$$

We assume that traffic is known in advance in both definitions, and take Y^{LTL} and Y^{HTL} as the maximum link load values observed during the LTL and HTL periods respectively. We compute two different robust routing configurations for both polytopes; RROP-LTL corresponds to the SRR configuration for polytope $\mathbb{X}^{\text{LTL}}$, and RROP-HTL for polytope $\mathbb{X}^{\text{HTL}}$. As we mentioned before, both RROP-LTL and RROP-HTL use the same set of paths, namely the paths obtained from Table 1 for polytope $\mathbb{X}^{\text{LTL}}$. Solving RROP for a given set of paths consists of only adding new extreme points of polytope $\mathbb{X}$ (i.e., only new constraints are added).

Results are presented in Fig. 3, which depicts (a) the maximum link utilization $u_{\max}$ and (b) the mean end-to-end queuing delay d_{mean} during the evaluation period. Let us first focus the attention on the performance of RROP-HTL after the 100th minute. Despite achieving an almost optimal performance as regards $u_{\max}$ (a relative difference with respect to the optimum smaller than 4%), RROP-HTL obtains a queuing delay that constantly exceeds the optimum by almost 40% under a moderate network load. Such a difference may not be even acceptable from a QoS perspective, where end-to-end delays are even more important than network congestion. As we will show later, this loss in performance is a direct consequence of the local criterion used in RROP.

A second interesting observation comes from the difference between RROP-HTL and RROP-LTL performances before and after the abrupt traffic volume increase; Fig. 3(a) shows that, despite an almost negligible network load, RROP-LTL outperforms RROP-HTL by almost 50% of relative utilization during the LTL period, while the opposite happens during the HTL period. The difference is not that big as regards delay before the 100th minute, but it becomes significant after the volume increase, where RROP-LTL obtains a very bad performance. These results are somehow expected given the polytopes definition, and brings to light both the dependence of RROP on the uncertainty set definition and the inherent consequence of using a single static routing configuration under large traffic variations. Let us highlight the fact that in this example we have considered that traffic was known in advance for the definition of both polytopes $\mathbb{X}^{\text{LTL}}$ and $\mathbb{X}^{\text{HTL}}$. While traffic during the LTL period is easy to predict, the definition of $\mathbb{X}^{\text{HTL}}$ in a real traffic scenario is a challenging task. We will come back to this issue in the following section.

Let us now discuss the results obtained by the dynamic schemes. A first important observation is that they present an important overshoot, with an absolute difference with the optimum $u_{\max}^{\text{opt}}$ of approximately 40%. Regarding MinDG in particular, convergence after the anomaly is very slow, taking more than 600 sec. However, it should be noted that when it eventually converges, it obtains a d_{mean} that is very similar to the optimum. In terms of $u_{\max}$, the difference with respect to the optimum is approximately 10%.

Special attention deserves the case of MinUG. After a shorter convergence time (approximately 100 sec.), the resulting value of $u_{\max}$ is not the optimum. Let us recall that this kind of game (which models schemes such as REPLEX of TeXCP) is used to converge to a routing configuration that minimizes the maximum link utilization [10, 11]. However, in this case, the difference is more than 15%. Both of these problems will be further discussed in the following section.

6. Improving the Algorithms Performance

The simple evaluation conducted in the previous section shows some conception drawbacks of the SRR and DLB algorithms presented in Sec. 3 and 4 respectively. In this section we shall explain the origin of these problems and present enhanced mechanisms to overcome them.

6.1. Improving Stable Robust Routing

6.1.1. Network-Wide Performance

As we showed in Fig. 3(b), the minimization of $u_{\max}$ leads to a distribution of traffic that results in an excessive end-to-end delay. Using the mean delay d_{mean} as the objective function in RROP (cf. Table 1) would be an interesting approach to ease the problem; however, $f_l(\rho_l)$ is a non-linear function and the optimization problem becomes too difficult to solve. As we previously said, optimization under uncertainty is more complex than classical optimization and simple optimization criteria should be used. In this sense, we could use instead the mean link utilization u_{mean} as the objective function.

The mean link utilization considers at the same time the load of every link in the network and not only the utilization of the most loaded link; as we will show in the results, such an objective function provides a better global

$$
\begin{array}{rcll}
\text{minimize} & u_{\text{mean}} & & \\
\text{subject to:} & & & \\
\sum_{l\in L}\sum_{k\in N}\sum_{p\in P_k}\frac{1}{c_l}\lambda_l^p\,.\,r_p^k\,.\,x(k) & \leqslant & u_{\text{mean}}.q & \forall\, X\in\mathbb{X} \\
\sum_{k\in N}\sum_{p\in P_k}\lambda_l^p\,.\,r_p^k\,.\,x(k) & \leqslant & u_{\max}^{\text{thres}}.c_l & \forall\, l\in L,\ \forall\, X\in\mathbb{X} \\
\sum_{p\in P(k)} r_p^k & = & 1 & \forall\, k\in N \\
r_p^k & \geqslant & 0 & \forall\, p\in P_k,\ \forall\, k\in N
\end{array}
$$

Table 2: Robust Routing Mean Utilization Optimization Problem (RRMP)

$$
\begin{array}{rcll}
\text{minimize} & u_{\text{aof}}=\alpha\,.\,u_{\max}+(1-\alpha)\,.\,u_{\text{mean}} & & \\
\text{subject to:} & & & \\
\sum_{l\in L}\sum_{k\in N}\sum_{p\in P_k}\frac{1}{c_l}\lambda_l^p\,.\,r_p^k\,.\,x(k) & \leqslant & u_{\text{mean}}.q & \forall\, X\in\mathbb{X} \\
\sum_{k\in N}\sum_{p\in P_k}\lambda_l^p\,.\,r_p^k\,.\,x(k) & \leqslant & u_{\max}.c_l & \forall\, l\in L,\ \forall\, X\in\mathbb{X} \\
\sum_{p\in P(k)} r_p^k & = & 1 & \forall\, k\in N \\
r_p^k & \geqslant & 0 & \forall\, p\in P_k,\ \forall\, k\in N
\end{array}
$$

Table 3: Robust Routing AOF Optimization Problem (RRAP)

performance as regards end-to-end delay. However, a direct minimization of u_{mean} does not assure a bounded maximum link utilization, which is not practical from an operational point of view. In this sense, we propose to change the objective function in RROP by u_{mean}, while bounding the maximum link utilization by a certain threshold $u_{\max}^{\text{thres}}$ defined a priori. The resulting problem, which we shall call the *Robust Routing Mean Utilization Optimization Problem* (RRMP), is defined in Table 2.

RRMP is solved in the same way as RROP, using the same recursive algorithm proposed in [6]. Note that the difference between the two problems is only a new constraint per each new traffic demand in $\mathbb{X}$ (in fact, for each extreme point of $\mathbb{X}$). The drawback of RRMP is its dependence on the value of $u_{\max}^{\text{thres}}$, which directly influences the routing performance as we will shortly see. An interesting choice for $u_{\max}^{\text{thres}}$ would be to use the output of RROP, namely $u_{\max}^{\text{robust}}$. To some extent this would result in a similar routing solution but with better traffic balancing.

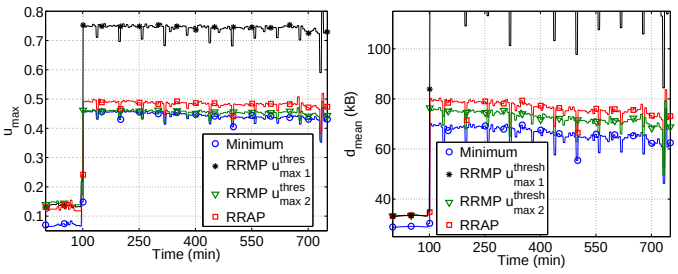
An alternative approach is to minimize both the value of $u_{\max}$ and u_{mean} at the same time, which constitutes a problem of multi-objective optimization (MOO). MOO problems are generally more difficult to solve because traditional single-objective optimization techniques cannot be directly applied. Nevertheless, the problem of finding all the Pareto-efficient solutions to a linear MOO problem is well known and different approaches can be used to treat the problem [32, 33]. In this work we consider an intuitive and easy approach to solve a MOO problem with standard single-objective optimization techniques. The approach consists in defining a single aggregated objective function (AOF) that combines both objective functions. We define a weighted linear combination of $u_{\max}$ and u_{mean} as the new objective function $u_{\text{aof}}=\alpha\,.\,u_{\max}+(1-\alpha)\,.\,u_{\text{mean}}$, where $0\leqslant\alpha\leqslant 1$ is the combination fraction. Despite its simple form, this new objective is very effective and provides accurate results for both performance indicators. We shall call this new optimization problem as *Robust Routing AOF Optimization Problem* (RRAP), defined in Table 3. As before, RRAP is solved with the same algorithms used in RROP.

6.1.2. Comparison between RRMP and RRAP

We will now evaluate both the RRMP and RRAP versions of SRR in the traffic scenario previously considered in Sec. 5. In order to appreciate the dependence of RRMP on the maximum link utilization threshold $u_{\max}^{\text{thres}}$, two different thresholds are used in the evaluation: $u_{\max_1}^{\text{thres}}=1$ (which corresponds to the constraint $u_{\max}\leqslant 1$ in Table 1), and $u_{\max_2}^{\text{thres}}=u_{\max}^{\text{robust}}$, where $u_{\max}^{\text{robust}}$ is the output of RROP-HTL in Sec. 5. In the case of RRAP, the weight α is set to 0.5, namely an even balance between $u_{\max}$ and u_{mean}. This may impress as a somewhat naive approach to the reader, but practice shows that this choice provides in fact very good results.

Figures 4 depicts the results in this case. Let us focus our attention on the operation after the 100th minute, as all robust routing configurations use $\mathbb{X}^{\text{HTL}}$ as the uncertainty set. To be as fair as possible, both RRMP and RRAP use the same set of paths as those used by RROP in Fig. 3. The figure clearly shows that the performance of RRMP strongly depends on the threshold $u_{\max}^{\text{thres}}$. In the case of $u_{\max_1}^{\text{thres}}$, the attained maximum link utilization is well beyond the optimal values, reaching almost a 70% of relative performance degradation. This overload directly translates into huge mean end-to-end queuing delays. Results are quite impressive when considering the second threshold, both as regards $u_{\max}$ and d_{mean}. RRMP using $u_{\max_2}^{\text{thres}}$ provides a highly efficient robust routing configuration, showing that it is possible to improve current implementations of SRR with a slight modification of the objective function. However, this dependence on the threshold $u_{\max}^{\text{thres}}$ introduces a new tunable parameter, something undesirable when looking for solutions that simplify network management.

As regards RRAP, obtained results are slightly worse than those obtained by RRMP $u_{\max_2}^{\text{thres}}$, but still very close to the optimal performance, with a relative performance degradation of about 10% as regards $u_{\max}$ and d_{mean} with respect to an optimal routing configuration. Nevertheless, RRAP has no tunable parameter apart from the combination factor α, which in fact is set to a half independently of the traffic situation.



(a) Maximum link utilization (b) Mean e2e queuing delay

Figure 4: Maximum link utilization and mean end-to-end queuing delay for RRMP and RRAP.

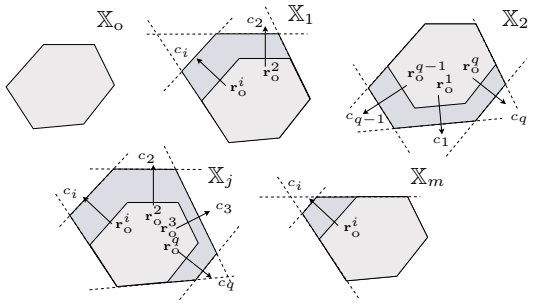


Figure 5: Different anomaly polytopes for preemptive robust routing computation.

6.1.3. The Reactive Robust Routing

As we showed in Sec. 5, the definition of the uncertainty set has a major impact on the performance of SRR. In particular, we saw that using a single definition of uncertainty set under highly variable traffic cannot provide routing efficiency for both normal operation traffic and unexpected traffic events. Despite being one of its most important features, using a single SRR configuration is not the best strategy.

In [13] we proposed an adaptive version of SRR, known as the *Reactive Robust Routing* (RRR). The basic idea in RRR consists in computing a primal robust routing configuration $R^{\mathrm{o}}_{\mathrm{robust}}$ for expected traffic variations in normal operation within a primal polytope $\mathbb{X}_{\mathrm{o}}$. This polytope is defined as in Sec. 3, based on a certain fixed routing configuration R_{o} and the expected links traffic load we shall call $Y_{\mathrm{o}} = \{\rho_{\mathrm{o}_i}\}$. Additionally, a set of m *anomaly* polytopes $\mathbb{X}_j$ are defined, and a preemptive robust routing configuration R^{j}_{robust} is computed for each of these anomaly polytopes.

Let us explain the concept of an anomaly polytope. In Fig. 3, the abrupt increase in traffic volume is caused by a single *anomalous* OD flow x_k that unexpectedly carries a many times bigger traffic load θ due to an external routing modification. After this exogenous unexpected event, the traffic demand X takes the value $X' = X + \theta.\boldsymbol{\delta}_k$, where $\boldsymbol{\delta}_k = (\delta_{1,k}, \ldots, \delta_{k,k}, \ldots, \delta_{m,k})^T$, $\delta_{i,k} = 0$ if $i \neq k$ and $\delta_{k,k} = 1$. We shall designate this unexpected traffic increase in OD flow x_k as anomalous traffic event A_k. The anomaly polytope $\mathbb{X}_k$ results from expanding the primal polytope $\mathbb{X}_{\mathrm{o}}$ in the directions of the links that traverses the anomalous OD flow x_k, with respect to R_{o}. The reader should bear in mind that the kind of unexpected traffic events we deal with are independent of the intradomain routing; these events originate outside the network and propagate between origin-destination nodes. This justifies the relevance of the polytope expansion with respect to R_{o}. The obtained polytope $\mathbb{X}_k$ is the smallest polytope that contains the unexpected traffic demand X' and thus, the corresponding robust routing configuration R^{k}_{robust} provides a relatively good performance under its occurrence. Figure 5 explains the idea of the multiple anomaly polytope expansion. As before, $\mathbf{r}^{i}_{\mathrm{o}}$ stands for the i-th row of the routing matrix R_{o}.

Note that in a real scenario it is not possible to predict the size of the anomalous traffic θ. As a consequence, the primal polytope $\mathbb{X}_{\mathrm{o}}$ is expanded to the limits of link capacities, obtaining the following anomaly polytope for each anomalous traffic event A_k:

$$\mathbb{X}_k = \left\{ X \in \mathbb{R}^m, R_{\mathrm{o}}.X \leqslant Y^{A_k},\ X \geqslant 0 \right\},\ \forall k \in N \qquad (6)$$

In (6), the i-th component of Y^{A_k} takes the value ρ_{o_i} if $r^{i,k}_{\mathrm{o}} = 0$, or the value c_i if $r^{i,k}_{\mathrm{o}} > 0$, being $r^{i,k}_{\mathrm{o}}$ the element (i,k) of R_{o}.

Given the primal and the m preemptive robust routing configurations $R^{\mathrm{o}}_{\mathrm{robust}}$ and R^{j}_{robust}, RRR uses an on-line anomaly detection/localization sequential algorithm to detect the occurrence of an anomalous event A_k, switching routing from $R^{\mathrm{o}}_{\mathrm{robust}}$ to R^{k}_{robust} (and from R^{k}_{robust} back to $R^{\mathrm{o}}_{\mathrm{robust}}$ when normal operation is regained). We refer the reader to [13] for additional details on the detection/localization algorithms and the implementation of RRR.

RRR can handle large and unexpected traffic variations in single OD flows quite effectively (the case of multiple simultaneous anomalies is beyond the scope of RRR). However, given the difficulty involved in modifying the routing configuration of a large scale network in an on-line fashion, the contributions of RRR are mainly theoretical. This problem can be solved by using a load balancing technique instead of a complete routing reconfiguration. In load balancing, we keep the same set of paths P_k for each OD pair k, and only modify the fractions of traffic sent along each path. Load balancing can be easily performed on-line and does not require any additional modifications in current path-based networks such as MPLS. We shall refer to the load balancing variant of RRR as *Reactive Robust Load Balancing* (RRLB), stressing the difference between routing reconfiguration and load balancing.

RRLB uses the same set of anomaly polytopes $\mathbb{X}_j$ defined in RRR, but the computation of the m preemptive robust routing configurations R^{j}_{robust} is slightly modified. The same set of paths P_k obtained during the computation of $R^{\mathrm{o}}_{\mathrm{robust}}$ is used in every R^{j}_{robust}. As in Sec. 5 and 6.1.2,

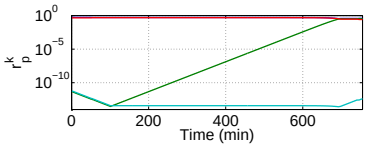


Figure 6: Evolution of r_p^k for the anomalous OD pair (MinDG)

routing configurations R^j_{robust} are obtained with a simplified version of the former optimization algorithm, where only new traffic demands are progressively added and no extra paths are created.

6.2. Improving Dynamic Load Balancing

6.2.1. Convergence Time

The DLB algorithms evaluated in Sec. 5 present an important overshoot and a significant settling time in the presence of sudden and large traffic variations. If the traffic anomaly is a perfect step, then the overshoot is unavoidable. We will try to address the long settling time instead. The reason behind this problem is relatively simple, as shown in Fig. 6. The graph depicts the evolution over time of the corresponding r_p^k values of the anomalous OD pair for MinDG in the example of Sec. 5. We may see that, although the r_p^k change exponentially fast, at the moment of the anomaly the values that should increase are so small that it takes them a very long time to converge. A possible solution is to impose a minimum value to all r_p^k. However, this will affect the precision of the algorithm and will still result in significant settling times.

Actually, r_p^k may be regarded as an indicator of the performance of path p in the previous iterations. A very small r_p^k means that p performed very badly with respect to the rest of the paths in the past. However, when the anomaly occurs, conditions severely change and history is no longer as relevant. If we consider that we are in such situation, we could for instance completely ignore history and restart the game by setting $r_p^k = 1/|P_k| \forall k \in P_k$. Before deciding how to reassign r_p^k, we will discuss how an OD pair may decide if it should restart its game or not.

Consider a situation where most of the traffic for OD pair k is routed along a path that is not the cheapest, and that the r_p^k corresponding to the minimum-cost path is very small. This could mean that although the former performed better in the past, this is no longer true and some traffic should be re-routed to the latter. This is more so as the difference in cost increases. However, this "suspicious" situation could be due to noisy measurements. To make sure that the game has actually changed and that it should be restarted, we will require such a situation to persists during a certain number of consecutive iterations. Once we detected that the game should be restarted, we will re-route some of the traffic that was being routed along the path with the biggest r_p^k to the cheapest one. The amount will be proportional to the relative difference in cost to avoid overreacting. Finally, remember that with WMA fast adaptation is achieved when the r_p^k are not too small. The objective with this "game restart" is simply to move r_p^k from critically small values. The algorithm will then rapidly converge to the optimum. We now present the pseudo-code of the complete algorithm for OD pair k:

Algorithm 2 WMA with Restart (WMA-R)

for $t = 1, \ldots, \infty$ **do**
  Obtain path costs $\phi_p \ \forall p \in P_k$
  Determine $p_{\min} = \underset{p \in P_k}{\operatorname{argmin}} \phi_p$ and $p_{\max} = \underset{p \in P_k}{\operatorname{argmax}} r_p^k$
  if $(r^k_{p_{\min}} < 0.1)$ and $(\phi_{p_{\min}} + \phi_{th} < \phi_{p_{\max}})$ **then**
    $n_e^k \leftarrow n_e^k + 1$
  else
    $n_e^k \leftarrow 0$
  end if
  if $n_e^k \leq n_{th}^k$ **then**
    Perform a normal iteration of WMA (cf. Algorithm 1)
  else
    $n_e^k \leftarrow 0$
    $\Delta_r \leftarrow \min\left\{\frac{\phi_{p_{\max}}}{\phi_{p_{\min}}} - 1, 1\right\} \times \frac{r^k_{p_{\max}} - r^k_{p_{\min}}}{2}$
    $r^k_{p_{\max}} \leftarrow r^k_{p_{\max}} - \Delta_r$
    $r^k_{p_{\min}} \leftarrow r^k_{p_{\min}} + \Delta_r$
  end if
end for

The new variable n_e^k counts the number of consecutive occurrences of a "suspicious" situation (we used $n_{th}^k = 3$). The threshold ϕ_{th} is to make sure that the difference in cost between paths is significant. In particular, for MinUG we used $\phi_{th} = 0.005$ and for MinDG $\phi_{th} = 0.2\phi_{p_{\min}}$. Finally, note that when the game is restarted, we re-route a certain amount of traffic from $p_{\max}$ to $p_{\min}$, but at most the amount of traffic routed along each path is equalized.

6.2.2. Converging to the Social Optimum in Bottleneck Games

In Sec. 5 we showed an example in which MinUG does not converge to the optimum, and obtains a difference of 15% with respect to the optimum MLU. The reason behind this poor performance is simply that MinUG does not take into account the result regarding the optimality of the WE and the *efficiency condition* discussed in Sec. 4.1 and originally presented in [15]. This result states that if at a WE all OD pairs send their traffic along paths with a minimum number of network bottleneck links (those with the maximum utilization in the whole network), the WE is optimal. The problem we analyze now is how to design a path cost function ϕ_p that takes into account this condition, so that when using it, the load-balancing algorithm converges, when possible, to the correct WE. Note that the condition is only sufficient, meaning that a WE that fulfills the efficiency condition may not exist. A simple example of such case is a single OD pair with two paths with different lengths, where all links have the same capacity. Anyhow, the two main difficulties in the design of such path cost are the following. Firstly, the number of bottleneck links in a path is an integer (thus not continuous on

r_p^k). Secondly, the probability of two links having exactly the same utilization is zero, and as such we should consider the number of links that have an utilization *similar* to the network bottleneck.

The objective is then to find a cost function that penalizes paths in which several links have similar utilizations (and that this utilization is the maximum in all the network), and that it does not switch between values to avoid oscillations. A candidate ϕ_p that fulfills these two conditions is the so-called *log-sum-exp* function. Consider a set of arbitrary numbers $A = \{a_i\}$, the log-sum-exp function $g(A)$ is defined as follows:

$$g(A) = \frac{1}{\gamma_A} \log \left(\sum_{i=1,\dots,|A|} e^{\gamma_A a_i} \right) =$$

$$a_{i^*} + \frac{1}{\gamma_A} \log \left(1 + \sum_{i=1,..,|A| \ \wedge \ i \neq i^*} e^{\gamma_A (a_i - a_{i^*})} \right) \quad (7)$$

Consider the special case in which $a_{i^*} = \max A$. It should be clear that if a_{i^*} is significantly bigger than the rest of the elements in A, the above convex and non-decreasing function constitutes an excellent approximation of a_{i^*}. In fact, it easy to prove that $a_{i^*} \leq g(A) \leq a_{i^*} + \log(|A|)/\gamma_A$, meaning that we may control the precision of the approximation through the parameter γ_A (the bigger this parameter, the more precise the resulting approximation). Moreover, as more elements in A are similar to the maximum, $g(A)$ approaches the upper bound, reaching it when all elements are the same.

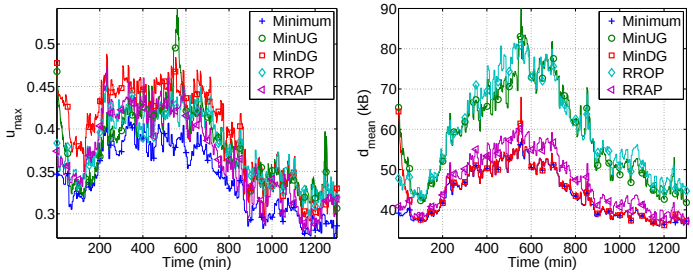
We will then use the second term of (7) as a penalty to those paths with several links whose utilization is similar to $u_{\max}$ (the maximum utilization in the network). More precisely, given a path p, let $U_p = \{u_l\}_{l \in p}$ be the utilizations in the path, and $l^* \in p$ be the link with the biggest utilization in p. We will then use the penalty function with the alternative set U_p^*, which has the same elements as U_p, but substitutes u_{l^*} by $u_{\max}$. This results in the following cost function:

$$\phi_p = u_{l^*} + \frac{1}{\gamma_p} \log \left(1 + \sum_{l \in p \ \wedge \ l \neq l^*} e^{\gamma_p (u_l - u_{\max})} \right) \quad (8)$$

Even if this new cost function penalizes paths with several network bottleneck links, it also penalizes longer paths, which was not our original objective. A good choice of γ_p will alleviate this side-effect. For instance, we used $\gamma_p = \log(|p|)/\max\{0.01, u_{l^*}/10\}$. This way, we try to minimize the effect of $\log(|p|)$ and relativize the penalization to u_{l^*}. The following section presents the results obtained by this new path cost function, which we will still call MinUG.

7. Evaluation and Discussion

In this section we evaluate the performance of the different RR and DLB algorithms presented in this work, considering both normal operation and anomalous traffic situations. We present and discuss three simulation case-scenarios: starting from a normal traffic variation scenario, we increase the number of OD pairs that present anomalous traffic variations. This allows for performance comparison at different levels of traffic variability. As both RRAP and RRMP provide similar results (when $u_{\max}^{\text{thres}}$ is correctly defined for RRMP), we will only consider the RRAP mechanism in the evaluation. Finally, we shall use RRLB-OP and RRLB-AP to designate the Reactive Routing Load Balancing variants of RROP and RRAP respectively.



(a) Maximum link utilization (b) Mean e2e queuing delay

Figure 7: Maximum link utilization and mean end-to-end queuing delay under normal operation.

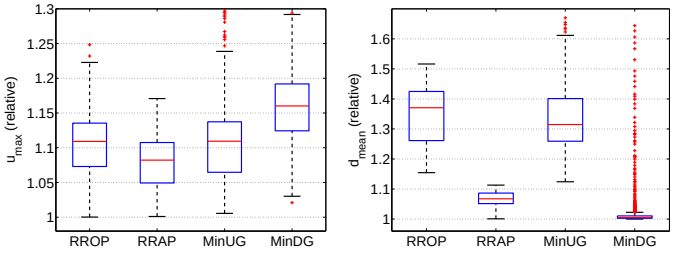
7.1. Normal Operation

The first case-scenario corresponds to traffic in normal operation. The only variability is due to typical daily fluctuations. Figure 7 presents the evolution of $u_{\max}$ and d_{mean} for RROP and RRAP, using a set of 260 TMs from dataset X01 in [29] (specifically, those with indexes between 420 and 680). All algorithms perform similarly as regards maximum link utilization, depicted in Fig. 7(a). This may be further appreciated in the boxplot summary presented in Fig. 8(a), where values are relative to those obtained with an optimal routing configuration. Note that the relative performance degradation is around 10% in most cases.

Figures 7(b) and 8(b) show that results are quite different as regards mean queuing delay. We may verify that the best results are obtained by MinDG, followed closely by RRAP. However, both RROP and MinUG systematically obtain a significant difference with respect to the optimum, generally between 30% and 40%. These results further highlight the limitations of RROP and MinUG as previously discussed: using $u_{\max}$ as a performance objective results in a relatively low maximum utilization, but neglects the rest of the links, impacting the network-wide performance.

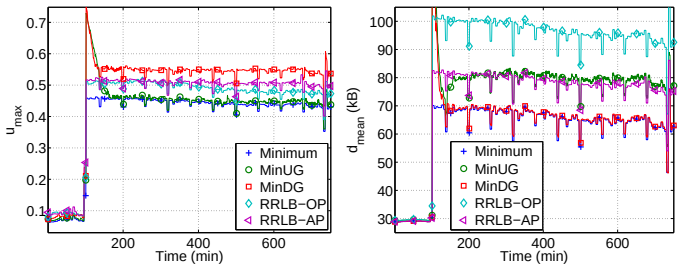
7.2. One Anomalous OD Pair

The second case-scenario is the one considered in Sec. 5, where there is a sudden and abrupt increase of the traffic



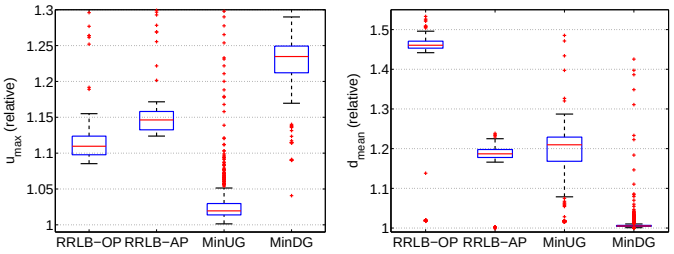
(a) Relative maximum link utilization (b) Relative e2e queuing delay

Figure 8: Maximum link utilization and mean end-to-end queuing delay under normal operation, boxplot performance summary. Depicted results are relative to the optimal values.



(a) Maximum link utilization (b) Mean e2e queuing delay

Figure 9: Maximum link utilization and mean end-to-end queuing delay under one anomalous OD pair.



(a) Relative maximum link utilization (b) Relative e2e queuing delay

Figure 10: Maximum link utilization and mean end-to-end queuing delay under one anomalous OD pair, boxplot performance summary. Depicted results are relative to the optimal values.

volume carried by one OD flow. As a difference with respect to the evaluation in Fig. 3, where traffic was assumed known in advance, this case-scenario corresponds to a real situation where traffic anomalies can not be forecast.

Firstly, notice in Fig. 9 how the improvements discussed in Sec. 6.2.1 for MinUG and MinDG result in a relatively smaller overshoot than before, but most importantly the settling time has been significantly decreased (in the case of MinDG, from 600 min. to less than 50 min). Moreover, note how the modified cost function proposed in Sec. 6.2.2 results in MinUG converging to the socially optimum WE.

Regarding $u_{\max}$, both RRLB-OP and RRLB-AP obtain similar results, with a relative performance degradation generally smaller than 15%. Note that while relatively important, this performance degradation is surprisingly small if we consider that traffic increases more than 500% in less than 10 minutes. The same may be said about MinDG, which obtains a degradation between 20% and 25%. In terms of d_{mean}, MinUG and RRLB-AP perform similarly. They both clearly outperform RRLB-OP, achieving a relative mean queuing delay almost 30% smaller. These results reinforces once again our observations about the difficulty in RROP to attain global performance, and the advantages of using a simple network-wide objective function in a robust routing algorithm. Moreover, they also illustrate the difference between MinUG and RROP. Even when MinUG was designed with the same objective than RROP (namely to minimize $u_{\max}$), the fact that in MinUG each OD pair greedily minimizes the path utilization results in a different overall behavior.

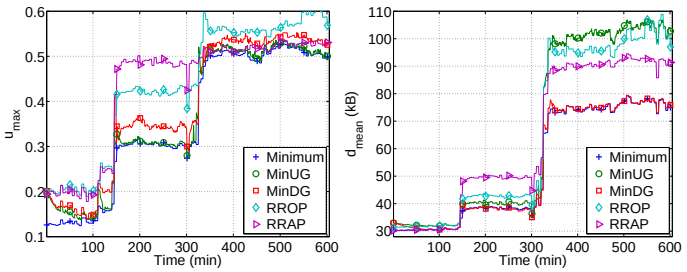
7.3. Two Anomalous OD Pairs

In this case-scenario two OD pairs largely increase their traffic demand, one at approximately the 150th minute and the other at the 320th (they correspond to TMs with indexes between 160 and 280 from dataset X06 in [29]). They both present this anomalous traffic until the end of the simulation. We shall then separate the simulation in three parts: the first third where traffic is normal, the second third were only one OD pair is anomalous, and the last third were both OD pairs are anomalous. The anomaly localization algorithm of RRR was designed for the case of one single anomalous OD pair. Because of this, we will further illustrate the tradeoff between size of the considered uncertainty set and efficiency of the obtained routing, and chose the uncertainty polytope by the traffic loads seen after the second anomaly.

In Fig. 11(a) we may see that, as expected, the $u_{\max}$ obtained by both RROP and RRAP in the last third of the simulation are very close to the optimum. However, in the rest of the simulation the difference may be important, specially in the second part where the absolute difference for RRAP is almost 20%. It is important to highlight the results obtained by MinDG and MinUG. Notice that the overshoot this time is much smaller than before (a maximum of 0.1 in $u_{\max}$ for MinDG) and the settling time is negligible. In this case, the increase in traffic of the anomalous OD pairs is more gradual than before, which clearly favors dynamic schemes in their performance.

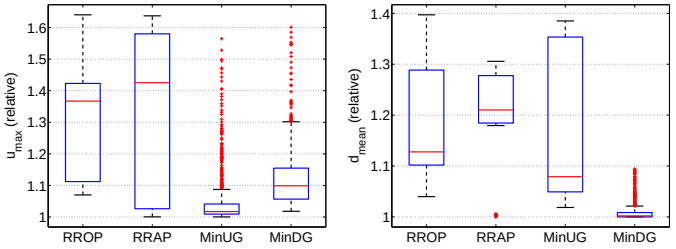
8. Conclusions and Future Work

From the study we presented in this article we may reach several conclusions. The most important is probably that we have shown that using a single routing configuration is not a cost-effective solution when traffic is relatively dynamic. Stable Robust Routing obtains a rather poor performance either when faced with non considered traffic demands (tight uncertainty sets) or when designed



(a) Maximum link utilization (b) Mean e2e queuing delay

Figure 11: Maximum link utilization and mean end-to-end queuing delay under two anomalous OD pair.



(a) Relative maximum link utilization (b) Relative e2e queuing delay

Figure 12: Maximum link utilization and mean end-to-end queuing delay under two anomalous OD pair, boxplot performance summary. Depicted results are relative to the optimal values.

to manage as many traffic demands as possible (big uncertainty sets). It is clear from our study that some form of dynamism is necessary, which could be either RRLB (Reactive Robust Load Balancing) or DLB (Dynamic Load-Balancing).

RRLB computes a nominal operation routing configuration, and has an alternative routing (using the same paths than in normal operation) for certain possible anomalous situations. In order to detect these anomalous situations, link load measurements have to be gathered [13]. On the other hand, DLB gathers these same measurements but also requires updating load-balancing in a relatively small time-scale. The added complexity is then to distribute these measurements to all ingress routers (instead of a central entity) and updating the load-balancing in real-time.

Our results show that the additional complexity involved in DLB is not justified when the variability (or the anomalies) are not very significant. However, the use of DLB under highly dynamic traffic is very appealing and generally provides better results than RRLB. Moreover, if the anomalies may not be correctly detected or localized (as in Sec. 7.3), the only effective solution is DLB.

Regarding RR in particular, we saw that using a local performance criterion such as the maximum link utilization (MLU) is not a suitable objective function as regards network-wide performance and QoS provisioning. In particular, we showed that an almost optimal robust routing configuration with respect to MLU can experience rather high mean end-to-end queuing delays, a very important performance indicator for all types of traffic. The maximum link utilization is widely used in current network optimization problems, particularly in most Robust Routing proposals, thus we believe that this simple evidence can help and should be considered in enhanced future implementations.

In fact, we have shown that objective optimization functions can be kept simple, and yet better network-wide performance can be attained. By using a simple combination of performance indicators such as the maximum and the mean link utilization, we obtained a robust routing configuration that definitely outperforms current implementations from a global end-to-end perspective, while achieving very similar results as regards worst-case link utilization.

The framework of Aggregated Objective Functions (AOF) we used provides interesting results as regards multi-objective optimization, particularly in the context of robust optimization. An AOF approach can be used to construct better objective functions from simple performance indicators, avoiding the need of more complex Multi-Objective Optimization (MOO) techniques. As part of our ongoing work we are currently analyzing the trade-off between using a simple AOF approach against a more complex but more complete MOO approach, computing all Pareto-efficient solutions for a polyhedral uncertainty set and comparing their performance.

In what respects DLB, dynamic approaches are generally met with reluctance due to their transient behavior under strong traffic variations. However, we have shown that this transient behavior can be effectively controlled, or at least alleviated, by simple mechanisms. Concerning the two different games we presented, conclusions are similar to those of RR. Striving to minimize d_{mean} instead of $u_{\max}$ results in a somewhat bigger maximum utilization, but a (sometimes much) better global performance.

It should also be highlighted that this article represents one of the first studies using no-regret algorithms for load-balancing, and as such much exploration is left to be done. In this sense, let us remark that WMA is arguably the simplest no-regret algorithm in the literature. For instance, it is easy to see that not all $\beta < 1$ guarantee a non-oscillatory behavior of the algorithm (consider for example $\beta = 0$). There exist other more sophisticated algorithms of this kind, that do not require any parametrization at all (including β) and still guarantee convergence (see for instance [34]), whose exploration is left for future work. However, an important conclusion of the current article is that, whatever the no-regret algorithm we choose, we will still require a "restart" feature as in WMA-R.

Our study also highlighted a problem with previously proposed DLB algorithms, namely the wrong assumption that OD pairs that greedily minimize the path utilization converge to a routing configuration that minimizes MLU. Based on a recent result [15], we have explored the pos-

sibility of modifying the path cost function so that the resulting routing configuration is actually optimum. Preliminary results presented in this article are very promising and encourage us to further study this new cost function (e.g. characterize the resulting WE or consider other alternative penalization functions).

9. Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was partially funded by CELTIC project TRANS and FP7 project Euro-NF.

[1] Cisco Systems, Global IP Traffic Forecast and Methodology 2006-2011, White Paper (2007 - updated 2008).

[2] Cisco Systems, The Exabyte Era, White Paper (2007 - updated 2008).

[3] Global Action Plan, An Inefficient Truth.
URL `http://www.globalactionplan.org.uk/upload/resource-/Full-report.pdf`

[4] M. Gupta, S. Singh, Greening of the Internet, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03), Karlsruhe, Germany, 2003, pp. 19–26.

[5] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, Deriving Traffic Demands for Operational IP Networks, in: IEEE/ACM Trans. on Networking, Vol. 9 (3), 2001, pp. 265–279.

[6] W. Ben-Ameur, H. Kerivin, Routing of Uncertain Traffic Demands, Opt. and Eng. 6 (3) (2005) 283–313.

[7] D. Applegate, E. Cohen, Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03), Karlsruhe, Germany, 2003, pp. 313–324.

[8] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, A. Greenberg, COPE: traffic engineering in dynamic networks, SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 99–110.

[9] A. Elwalid, C. Jin, S. Low, I. Widjaja, MATE: MPLS adaptive traffic engineering, in: IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM 2001), Vol. 3, Anchorage, USA, 2001, pp. 1300–1309.

[10] S. Kandula, D. Katabi, B. Davie, A. Charny, Walking the tightrope: responsive yet stable traffic engineering, in: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '05), Philadelphia, USA, 2005, pp. 253–264.

[11] S. Fischer, N. Kammenhuber, A. Feldmann, REPLEX: dynamic traffic engineering based on wardrop routing policies, in: Proceedings of the 2006 ACM CoNEXT conference (CoNEXT '06), Lisboa, Portugal, 2006, pp. 1–12.

[12] A. Khanna, J. Zinky, The revised ARPANET routing metric, SIGCOMM Comput. Commun. Rev. 19 (4) (1989) 45–56.

[13] P. Casas, L. Fillatre, S. Vaton, Robust and Reactive Traffic Engineering for Dynamic Traffic Demands, in: Next Generation Internet Networks (NGI 2008), Krakow, Poland, 2008, pp. 69–76.

[14] A. Blum, E. Even-Dar, K. Ligett, Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games, in: Twenty-Fifth Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC 2006), Denver, USA, 2006, pp. 45–52.

[15] R. Banner, A. Orda, Bottleneck Routing Games in Communication Networks, IEEE Journal on Selected Areas in Communications 25 (6) (2007) 1173–1179.

[16] M. Roughan, M. Thorup, Y. Zhang, Traffic engineering with estimated traffic matrices, in: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC '03), Miami Beach, USA, 2003, pp. 248–258.

[17] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, D. Towsley, On optimal routing with multiple traffic matrices, in: 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Vol. 1, Miami Beach, USA, 2005, pp. 607–618.

[18] P. Casas, S. Vaton, An Adaptive Multi Temporal Approach for Robust Routing, in: Euro-FGI Workshop on IP QoS and Traffic Control, Lisbon, Portugal, 2007.

[19] F. Larroca, J.-L. Rougier, Minimum-Delay Load-Balancing Through Non-Parametric Regression, in: Proceedings of the 8th International IFIP-TC 6 Networking Conference (NETWORKING '09), Aachen, Germany, 2009, pp. 782–794.

[20] M. Johansson, A. Gunnar, Data-driven traffic engineering: techniques, experiences and challenges, in: 3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS 2006), San José, USA, 2006, pp. 1–10.

[21] I. Juva, Robust Load Balancing, in: IEEE Global Telecommunications Conference (GLOBECOM '07), Washington D.C., USA, 2007, pp. 2708–2713.

[22] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.

[23] D. Mitra, K. Ramakrishnan, A case study of multiservice, multipriority traffic engineering design for data networks, in: 1999 Global Telecommunications Conference (GLOBECOM '99), Vol. 1B, 1999, pp. 1077–1083.

[24] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, L. Wynter, A survey on networking games in telecommunications, Comput. Oper. Res. 33 (2) (2006) 286–311.

[25] F. Larroca, J.-L. Rougier, Routing Games for Traffic Engineering, in: IEEE International Conference on Communications (ICC '09), Dresden, Germany, 2009.

[26] J. Wardrop, Some theoretical aspects of road traffic research, Proceedings of the Institution of Civil Engineers, Part II 1 (36) (1952) 352–362.

[27] R. Yaroshinsky, R. El-Yaniv, S. S. Seiden, How to Better Use Expert Advice, Mach. Learn. 55 (3) (2004) 271–309.

[28] N. Littlestone, M. K. Warmuth, The weighted majority algorithm, Inf. Comput. 108 (2) (1994) 212–261.

[29] Yin Zhang, Abilene Dataset.
URL `http://www.cs.utexas.edu/ yzhang/research/AbileneTM/`

[30] The Abilene Observatory.
URL `http://abilene.internet2.edu/observatory/`

[31] K. Cho, WIDE-TRANSIT 150 Megabit Ethernet Trace 2008-03-18.
URL `http://mawi.wide.ad.jp/mawi/samplepoint-F/20080318/`

[32] J. Evans, R. Steuer, A Revised Simplex Method For Linear Multiple Objective Programs, Mathematical Programming 5 (1) (1973) 54–72.

[33] J. Ecker, I. Kouada, Finding All Efficient Extreme Points for Multiple Objective Linear Programming Programs, Mathematical Programming 14 (1978) 249–261.

[34] P. Auer, N. Cesa-Bianchi, C. Gentile, Adaptive and Self-Confident On-Line Learning Algorithms, Journal of Computer and System Sciences, vol. 64, 2002, 48–75.

INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2012; **22**: 162–179
Published online 14 September 2011 in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/nem.799

Maximum delay computation for interdomain path selection

Isabel Amigo[1,2,*†], Sandrine Vaton[2], Thierry Chonavel[3] and Federico Larroca[1]

[1]*Facultad de Ingeniería, Instituto de Ingeniería Eléctrica, Universidad de la República, Montevideo, Uruguay*
[2]*Département Informatique, Telecom Bretagne, Brest, France*
[3]*Département Signal et Communications, Telecom Bretagne, Brest, France*

SUMMARY

One important problem when deploying interdomain path selection is advertising metrics that hold for a long period of time. In this paper we propose a method to aid interdomain path selection mechanisms in that sense. We present a means of computing a bound on the end-to-end delay of traversing a domain considering that the traffic varies within a given uncertainty set. This provides a robust and a verifiable quality of service value for traversing the autonomous system (AS), without revealing confidential information. Consequently, the bound can be safely conceived as a metric to be announced by each AS in the process of interdomain path selection. We show how the maximum delay value is obtained for an interdomain bandwidth demand and we propose an exact method and a numerical approximation method for computing it, neither of which rely on a complex monitoring infrastructure. Simulations with real data that illustrate the problem and validate our results are also presented. Copyright © 2011 John Wiley & Sons, Ltd.

Received 16 December 2010; Revised 10 June 2011; Accepted 17 June 2011

1. INTRODUCTION

There is increasing interest in value-added services such as videoconferencing or other bandwidth-on-demand services. In this context quality of service (QoS) and how to guarantee it become a crucial issue for all involved actors, i.e. the network provider, the service provider and the customer. This is especially difficult when the service traverses several domains or autonomous systems (AS). In this case, QoS must be provided by all the ASs involved, which raises several technical, economic and political issues. Concerning the technical aspects, achieving scalability, preserving confidentiality and providing interoperability is paramount in any solution [1].

We will focus on point-to-point services with QoS requirements. In this case the service may be abstracted to a QoS guaranteed tunnel (for instance, a Multiprotocol Label Switching (MPLS) tunnel [2]). The path must cross those domains through which destination is reached and whose combination of QoS parameters fulfills the service requirements.

In the framework of an alliance of ASs, carriers are envisioned to work together in order to achieve a common interest. In this scenario QoS values related to each domain are exchanged, and traffic engineering decisions are taken afterwards. Different mechanisms have been proposed for the selection and establishment of interdomain QoS-constrained tunnels, which mainly rely on the Resource Reservation Protocol—Traffic Engineering (RSVP-TE) [3] and path computation element (PCE) architecture [4–8]. These mechanisms are based on metrics announced by each AS but they do not specify how to compute such metrics. In any case, the announced metrics have to hold for some period of time, ideally as long as the service is provided. Hence it would be interesting for ASs to be able to provide QoS values that are guaranteed to hold for a certain period of time.

*Correspondence to: Isabel Amigo, Facultad de Ingeniería, Instituto de Ingeniería Eléctrica, Universidad de la República, Montevideo, Uruguay.
†E-mail: iamigo@fing.edu.uy

Other approaches providing methods for end-to-end QoS can be found in the literature. For instance, some propose extensions to the de facto standard interdomain routing protocol BGP (Border Gateway Protocol) [9]. Others propose ad hoc functions to BGP [10,11]. These are based on self-adaptive methods and perform routing decisions at the edge routers level in order to maintain certain QoS parameters below some given bounds. They monitor the network state obtaining feedback which acts as an input to the self-adaptive engine. These methods are conceived to work in a pure BGP network. However, we are interested in the case of explicitly signaled tunnels, like the PCE-based mechanisms, since they are more suitable in the context we are working on. For instance, we seek a method that strictly achieves the QoS needed and not only soft QoS. In addition, for reasons explained below, we seek a method with light dependence on monitoring.

For certain services, available bandwidth and end-to-end delay are critical parameters. The latter is composed of the sum of the delays introduced by each transit AS and the terminal ones, from source to destination. As illustrated in Figure 1, where we show a situation with two terminal ASs and one transit AS, the delay in each of the ASs depends on the traffic already present in the AS (t_* flows in Figure 1), the topology, the routing configuration, and the traffic coming from the new tunnel (flow u in Figure 1).

Naively, we may think that the problem of choosing the delay value to advertise can be reduced to simply advertising the current one. However, this presents two main problems, as commented in the following.

Monitoring the delay is itself a complex task. Several techniques have been proposed in the literature, mainly based on passive measurements, where some packets are timestamped and sent to a collector [12,13], or on active ones, where probe packets are sent along the network and the delay is inferred from the one experienced by the probe packets [14,15]. These techniques present several drawbacks. Just to mention the most common of them, they usually present issues of bandwidth consumption and the need for synchronization, for instance, based on specific equipment such as GPS devices. Moreover, all techniques need a monitoring architecture, which can become complex when accuracy is needed [16].

In addition, even if we were able to accurately measure the delay, the announced value, as mentioned above, should hold for a certain period of time. In this scenario the complexity is mainly due to the existence of uncertainty. This uncertainty can be classified into two types: network state uncertainty and traffic uncertainty.

Uncertainty in network state refers to the situation where the topology changes or is partially known. This may be due to information arriving out of date or not synchronized to the entity performing the computation, or simply to link failures. In the literature some approaches have been proposed for performing QoS routing under this kind of uncertainty [17–19]. However, in the present paper we will assume that the topology does not change, and considering this uncertainty is left for future work.

On the other hand, we will consider uncertainty in the traffic. This refers to the fact that the flows traversing the domain are not perfectly known. Knowing exactly the volumes of these

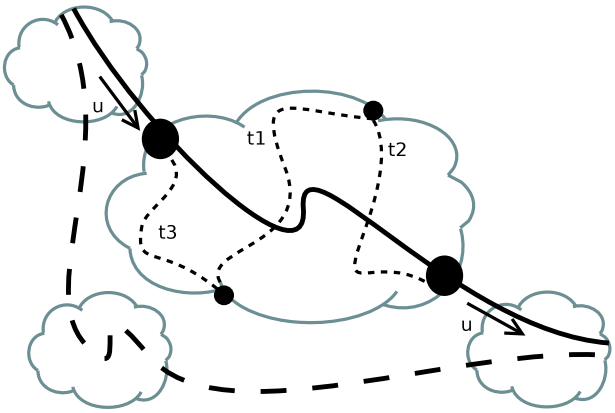


Figure 1. Scenario

flows, which we shall call Origin Destination traffic flows, requires a measurement infrastructure that is not always present, or could be expensive to implement. Techniques based on flow-level measurements, like Netflow [20], are very expensive for routers in terms of computational cost, while their sample-based version can lead to errors in traffic volume estimation. Techniques based on Simple Network Management Protocol (SNMP) data considerably reduce the CPU load on routers. In that case, the measured data consist of volumes of traffic traversing the different links of the network. In order to estimate the Origin Destination traffic volumes an ill-posed linear inverse problem has to be solved. Several methods exist in the literature for doing so [21–23]. Moreover, traffic uncertainty is not only related to the complexity on measuring the Origin Destination flows, but also to the fact that traffic may change rather frequently. There can be several reasons for these changes; for instance, external routing modification, the presence of unexpected events such as network equipment failures outside the domain, large-volume network attacks or flash crowd occurrences [24].

In summary, we aim at finding a valid end-to-end QoS metric. Thus two approaches could compete. Either we follow a dynamic approach, in which network state is continuously monitored and the metric value is updated, or we use a robust approach, in which a bound for the metric is provided. Reactive approaches make it possible to tightly follow the variations of the traffic but they require a monitoring infrastructure to be present and some sophisticated algorithms to process the measurement data. Moreover, reactive approaches are able to detect variations in the traffic demand such as abrupt changes but they are not able to forecast them [25]. On the contrary, proactive mechanisms provide pessimistic values of QoS metrics but they are able to provide metrics values which should hold for a given period of time since in that case uncertainty is taken proactively into account.

In this work we employ the proactive approach and consider the situation where traffic variation is the principal cause of delay variation. Thus we shall focus on the computation of a bound for the end-to-end delay of traversing an AS through a particular path as a function of the AS parameters we mentioned before: the routing configuration, the traffic demands and the traffic injected through the new tunnel. We assume that the topology and the routing configuration are fixed. However, we consider that traffic is non-static, and that it is contained in a so-called uncertainty set [26]. The question of how to choose this set is discussed later in the paper.

In this context, we provide an exact method and an approximate solution for solving the problem, which renders a solution arbitrarily close to the exact solution and lower computation complexity. These solutions do not require any complex monitoring infrastructure to be deployed. The solutions are tested using topology and data from real networks.

The obtained value can be afterwards used to advertise in the context of ASs path selection, since it is a QoS parameter bound and does not introduce confidentiality vulnerabilities. The latter refers to the fact that no topology information is delivered, but just the delay of traversing the AS, where the AS is seen from the outside as a black box.

The remainder of this paper is organized as follows. Section 2 introduces the assumptions and notation and formally states the problem. In Section 3 we show an exact solution to the problem and evaluate it through simulations. Section 4 presents an approximate solution with lower computational complexity than the exact one. Section 5 shows numerical results. Finally, conclusions and guidelines for future work are given in Section 6.

2. PROBLEM STATEMENT

In this section we formally present the problem of finding the maximum end-to-end delay experienced by a bounded amount of traffic traversing an AS through a particular path. As mentioned before, we will consider that traffic varies within an uncertainty set. First, let us introduce the notation that will be used throughout the paper and state some assumptions.

2.1. Assumptions and notation

The network is compounded of n nodes and of a set L of links, $L=\{l_1 \ldots l_{|L|}\}$, where the notation $|\cdot|$ refers to the cardinality of the set. Traffic demands will be represented by the so-called traffic matrix

DOI: 10.1002/nem

TM=$\{tm_{i,j}\}$, where $tm_{i,j}$ is the mean amount of traffic from node i to node j. We shall also use the term Origin Destination (OD) flows to refer to them. We reorder every traffic demand and rewrite the OD flows ($tm_{i,j}$) in vector form as t, $t=\{t_k\}$, $k=1\ldots n(n-1)$. The amount of traffic coming from the interdomain injected into the new tunnel will be u.

The link load $Y=\{y_i\}_{i\in L}$ is a vector containing in the i-th entry the load on link l_i without considering u. With these definitions we can see that $Y=R.t$, where R, a $|L|\times m$ matrix ($m=n(n-1)$), is the routing matrix, which means that $R_{i,j}=1$ if flow j traverses link i, and 0 otherwise.

The flow that carries u will traverse the AS from an origin to a destination node following a certain path. We will call this path P. We will equally refer to the set of links that belong to that path as P; in this case titP is a subset of L.

The mean link delay is approximated by the M/M/1 model, i.e. $D_l = \frac{K}{c_l - y_l}$, where c_l is the capacity of the link l and K the mean packet size. We then obtain the delay of a path as the sum of the mean delay of the links it traverses:

$$\text{Delay}_P = \sum_{l\in P} \frac{K}{c_l - y_l} \tag{1}$$

The propagation delay may be ignored in our formulation since it does not change with the load and may be added as a constant later on. Moreover, the M/M/1 model is used for illustrating the procedures towards a solution. In fact, any convex function may be used instead. For instance, the interested reader should consult Larroca and Rougier [27] for a method to obtain a good convex approximation of the delay function based on measurements. The same procedures explained in this paper shouldthen be repeated but with the new function. We will ignore the constant K in the following formulations, for the sake of notation simplification.

2.2. *Traffic demands*

As mentioned above, we will not make any assumptions on the traffic matrix except that it always belongs to a certain uncertainty set. In particular, we will follow the approach presented in Ben-Ameur and Kerivin[26] and define the uncertainty set as a polytope formed by the result of the intersection of several half-spaces. Consequently, all constraints can be written as $At\leq b$, where A is a certain matrix that can be defined according to different models, and titb is a given bound.

One such polytope was considered in the so-called Hose model [28] in the context of VPN services specification. This model establishes bounds in the ingress and egress points of a network. It is suitable for the case of VPN, where the ingress and egress values are easily known, but no detailed information regarding the network is available. However, in the context of interdomain path selection, we would like to have a smaller polytope, which is obtained with more detailed information, which would allow us to have a tighter bound.

An alternative is the *links capacity model*. This model results from the application of bounds on the total traffic traversing the different links of the network. Its definition can vary from a simple static one, imposing the physical constraints,i.e. links capacity, to a more dynamic one, allowing the constraints to be obtained from historical metrics. In the latter, the constraints can be written as $R^h t\leq b$, where $b=\{b_i\}$ is the vector of a historical link load and R^h is the routing matrix at the moment when the measurements were taken. This approach is used, for example, in Johansson [29], where a polyhedral definition of the traffic matrix is preferred to its estimation because of non-stationarity artifacts and estimation errors.

The links capacity model with historical bounds, for instance considering the maximum observed link load, provides more detailed information than the Hose model, along with dynamism, while it is still simple to obtain. The polytope can be frequently updated but does not require complexity for its computation.

Yet another alternative for computing tighter polytopes are prediction-based mechanisms. In this case the polytope is defined through imposing bounds on the value of traffic demands which are based on traffic prediction. The prediction of future demandsis based on past observations. For example

DOI: 10.1002/nem

artificial intelligence methods such as neural networks or time series analysis can be used in order to forecast future values of the traffic demand; see, for example, Fillatre *et al*. [30] for prediction based on a seasonal autoregressive integrated moving average (ARIMA) model. These mechanisms provide a more dynamic polytope, which must be updated according to predictions but involves more complexity. The result is a tighter polytope that provides, in turn, a tighter bound.

The choice of model for defining the polytope involves a trade-off between complexity and tightness of the bound. As we have shown above, simpler approaches could be used providing looser bounds, or more complex ones, needing in addition to be updated frequently, to provide tighter bounds. In the remainder of this work we shall use the links capacity model, though the solutions provided are still valid for any other model. We shall consider historical maxima for the bounds; thus measurements have tobe carried out. These measurements can be performed using SNMP, which is a widely deployed protocol. Since the value needed is just the overall interface traffic volume, we can safely assume that these values will be available on any AS.

2.3. Mathematical formulation

For the path traversed by the new tunnel the maximum link delay will be computed allowing the flows t to vary within a polytope. Therefore, we will work with a maximization problem with linear constraints. In order to have a more compact notation of the problem we shall define the m-dimensional column vector w_l, $l \in P$, as $w_l = \{w_{l,i}\} = R_{l,i}/c_l$.

The optimization problem is described by equation (2), where A and b define the polytope:

$$\begin{aligned} \max_{t} \quad & \sum_{l \in P} 1/c_l \frac{1}{1 - w_l^T t - u/c_l} \\ \text{s.t.} \quad & At - b \leq 0 \end{aligned} \tag{2}$$

Please note that if some additional linear constraints must be taken into account they can be integrated into the definition of the polytope $At \leq b$. One example of such constraints can be $w_l^T t + u/c_l < 1$, for $l \in P$, which simply states that thereshould be enough link capacity in order to accommodate all the traffic, including the new tunnel.

We can see that the objective function in the maximization problem (2) is not a concave function; consequently, the problem is not a convex one. On the contrary, the problem is the maximization of a convex function over a polytope. This is a very difficult problem—all the more so since the objective function is not strictly convex (as shown in Appendix A).

Intuitively we can see that the reason why the function is not strictly convex is the difference between the number of links and the number of OD flows. Indeed, while the number of links grows linearly with the number of nodes in the network, the number of OD flows squares with the number of nodes in the network. This means that for different values of the vector t the objective function of problem (2) can have the same value, while its gradient always remains non-negative. In the following section we reformulate the problem and show a possible way to find its solution.

3. FINDING THE EXACT SOLUTION

3.1. Formulation

We now state the problem in a different way and show a method for finding the exact solution. We aim at formulating the problem in such a way that the objective function is strictly convex and the dimension of the problem is reduced. For doing so we shall decompose the vector t over a particular basis of R^m.

The procedure consists in decomposing the vector t over the vectors w_l, $l \in P$, and their orthogonal complement. We define the matrix W_1 as an m by $|P|$ matrix, whose columns are the vectors w_l, with $l \in P$, and W_2, an m by $m - |P|$ full rank matrix such that it verifies

DOI: 10.1002/nem

$$W_1^T W_2 = 0 \tag{3}$$

In other words, the columns of W_2 form the basis of the space orthogonal to the one spanned by the columns of W_1.

Provided that the columns of W_1 are also linearly independent, the columns of the matrix W defined after W_1 and W_2 as

$$W = [W_1 W_2] = \left[w_1, \ldots, w_l, \ldots, w_{|P|}, w_m\right] \tag{4}$$

represent a basis of R^m.

We shall decompose the vector t over the defined basis using the auxiliary variables $x \in R^{|P|}$ and $h \in R^{m-|P|}$ as

$$t = W_1 x + W_2 h \tag{5}$$

By multiplying both sides of equation (5) by w_l^T and using equation (3) we obtain

$$w_l^T t = w_l^T W_1 x = v_l^T x \tag{6}$$

where we have set $v_l^T = w_l^T W_1$, for all $l \in P$. Note that both v_l and x are column vectors of dimension $|P|$.

Equation (6) will directly lead us to rewriting the objective function of problem (2) as a function of x. We shall now redefine the polytope by writing it in the basis W. For doing so, the change of variables defined by equation (5) needs to be done under the constraints of problem (2). This leads to defining a new matrix denoted D and computed as AW. The polytope over the new basis can be compactly written as $D[x^T h^T]^T \leq b$.

All in all, problem (2) can be rewritten in the form of problem (7). Note that the objective function depends only on the variable x:

$$\begin{aligned} \max_x \quad & \sum_{l \in P} 1/c_l \frac{1}{1 - v_l^T x - u/c_l} \\ \text{s.t.} \quad & \mathrm{D}\begin{pmatrix} x \\ h \end{pmatrix} \leq b \end{aligned} \tag{7}$$

Let us refer to the objective function of problem (7) as $J(x)$ and the new polytope as V (i.e. $V = \{[x^T h^T]^T \in R^m : D[x^T h^T]^T \leq b\}$). Let us as well define the polytope V_x as

$$V_x = \left\{x \in R^{|P|} \middle| \exists h \in R^{m-|P|} : D\left[x^T h^T\right]^T \leq b\right\} \tag{8}$$

Let $W_1 = \text{span}\{w_1 \ldots w_{|P|}\}$, where 'span' refers to the set of all linear combinations of vectors $w_1 \ldots w_{|P|}$. Clearly V_x is the projection of V onto W_1.

Since V is a convex polytope by definition, it is easy to check that V_x is also a convex polytope. More precisely, V_x is the convex hull of the projection of the extreme points of V onto W_1 [31].

Then, since $J(x)$ does not depend on h, problem (7) can be represented in the space W_1 as follows:

$$\begin{aligned} \max_x \quad & J(x) \\ \text{s.t.} \quad & \mathrm{x} \in V_x \end{aligned} \tag{9}$$

It is proven in Appendix B that $J(x)$ is strictly convex in V_x and that the solution of problem (10) is attained at an extreme point of the polytope V_x.

Problem (10) allows us to work with a strictly convex function, and to reduce the dimension of the feasible region—in some cases, considerably. In order to find the solution, we need to be able to perform the projection of a polytope, and afterwards enumerate its extreme points. Methods for doing so

DOI: 10.1002/nem

are available [32], although these can be computationally expensive tasks. In the following subsection we explore this solution by performing simulations in a real topology.

3.2. Simulations

In order to assess the results of the proposed method we will use the Abilene Network, whose topology, historical traffic demands and routing matrix are available from Zhang [33]. The Abilene Network consists of 30 internal links and 12 routers (all exchanging traffic among them).

For defining the polytope, linear constraints were set using the links capacity model with a maximum in the link utilization equal to the maximum achieved by historical data. We actually did a scaling of these data in order to obtain a more interesting case, since historically link utilizations in the Abilene Network have been quite small.

Figure 2 shows a traffic trace of the Abilene Network. This example shows how the traffic matrix is prone to sudden traffic variations. Figure 2(a) shows the traffic for some OD flows corresponding to 2016 consecutive measurements (where each color corresponds to one OD flow), while Figure 2(b) shows the link load (each color corresponds to the load of a certain link).

In order to test the method in different scenarios, we compute the solution for four paths between a pair of origin and destination nodes and three different amount of bandwidth for the interdomain request. The paths were computed using a shortest path algorithm, and keeping the four shortest paths. Note that the method of selecting the paths could have been a different one. This choice and its impact on the delay are beyond the scope of the present paper. The values for the bandwidth of the interdomain demands (i.e. u) are 1, 19.4 and 270 Mbps. These values could correspond, for instance, to a VoIP service, a broadcast quality HDTV service and a VPN service, respectively.

For each path we will compute the maximum delay suffered by a flow traversing it and carrying each one of these services at a time. All results were computed on a recent machine with good computational power (two Intel Xeon X5660 processors, 2.80 GHz, 24 GB of RAM). For computing the polytope projection and enumerating its extreme points we used the MPT library [34].

In Figure 3(a) a comparison of the maximum delay value for the fourth shortest path and the real mean delay of the same path at a certain instant is shown. For this particular case the maximum delay value is approximately three times more than the instantaneous one. This highlights the need for announcing a value that the domain is able to assure, rather than a measured one. Figure 3(b) shows the same bound, compared to several delay values occurring during the two weeks after computing the bound. Again, this shows that the delay can vary considerably and considering a value obtained in a given moment could not be safe. The bound in this case is quite loose; it should be noted that it could be tightened by including more information in the description of the polytope, for instance, considering the Hose model at the same time, as introduced in Section 2.2.

The time consumed in performing the computation of all demands for one path varied between 4 and 38 min, which for a moderately sized network is rather high. In fact, even if in several topologies we were able to find the exact solution through these means, it is still an open question whether there exists an algorithm for enumerating all extreme points of a polytope of an arbitrary dimension in polynomial running time [35]. In the following section, we present an approximate method that can be used as an alternative to the previous one when its computational time becomes excessive.

4. FINDING AN APPROXIMATE SOLUTION

In Section 3 we have presented a method that allows to find the exact solution. Nevertheless, we have pointed out that its complexity remains open. In this sense, we now present a method that provides an approximate solution to problem (2), while reducing computational time. More precisely, we present a numerical method based on the approximation of the objective function by a piecewise linear function. This method provides a value that is arbitrarily close to the exact solution (up to some controlled error).

Let us introduce the method with a detailed description of the procedure to obtain it. First, we transform each link's delay function, $\frac{K}{c_l}\frac{1}{1-y_l/c_l-u/c_l}$, into a piecewise linear function over y_l. For this, we partition each function's domain into η_l subintervals and approximate the function in each subinterval by its first-order Taylor polynomial. We shall note the subintervals of link l as $\Delta_{l,j}$, $j=1\ldots\eta_l$.

DOI: 10.1002/nem

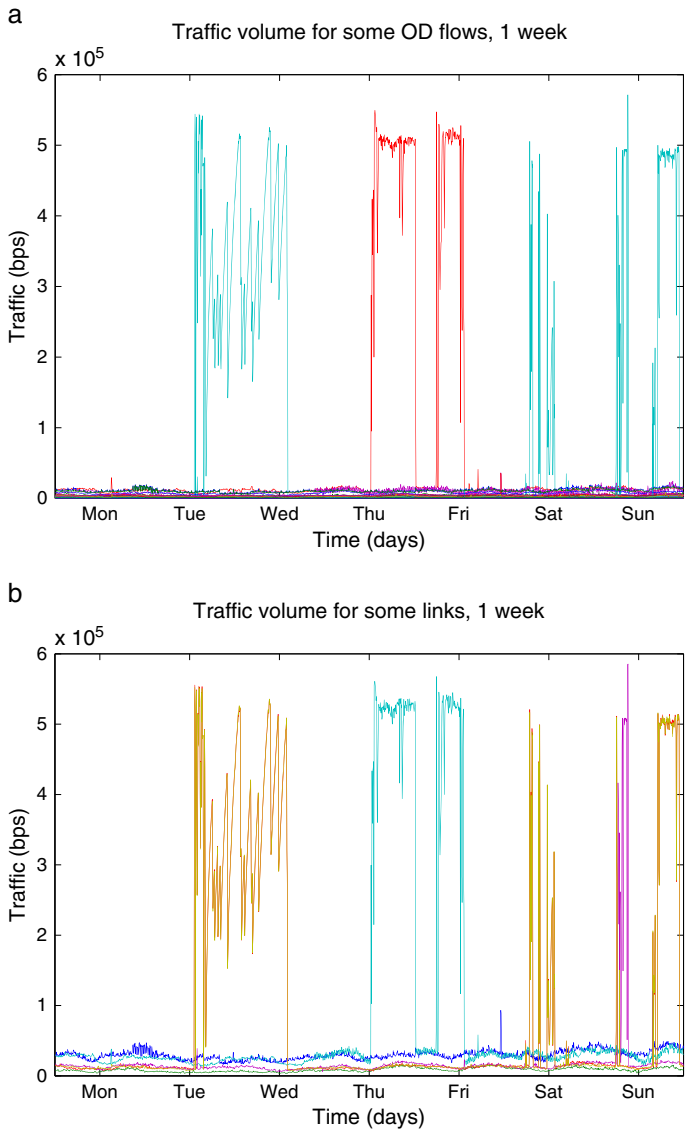


Figure 2. Example of traffic variation in the Abilene Network—one week of traffic. (a) Traffic volume per OD flow. (b) Link load

Secondly, we obtain the delay of the path, as before, by summing the delay on each link belonging to it. Therefore, we obtain a maximization problem similar to (2) but now with a piecewise linear objective function. Let us utilize the indicator function, defined as

$$1_{\Delta}(x) = \begin{cases} 1 & \text{if } x \in \Delta \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The new problem can then be seen in (11), where α and β are taken from the Taylor polynomial of the original function:

$$\max_{t} \quad \sum_{l \in P} \sum_{j=1}^{\eta_l} \left(\alpha_{l,j} w_l^T t + \beta_{l,j}\right) 1_{\Delta_{l,j}}\left(w_l^T t\right) \tag{11}$$

$$\text{s.t.} \quad At - b \leq 0$$

DOI: 10.1002/nem

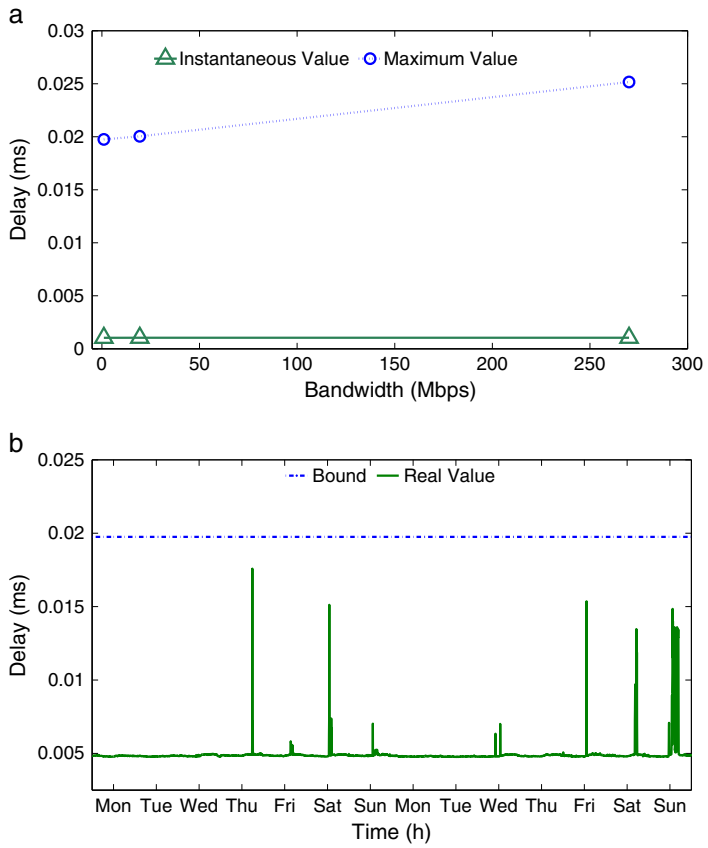


Figure 3. Simulations on the Abilene Network. (a) Comparison of the maximum delay value and an instantaneous one-for-one path. (b) Comparison of the real delay values and a computed bound for one path and one bandwidth demand

The next step is to redefine problem (12) in order to eliminate the indicator function and to obtain a linear objective function. To do so, we decompose problem (12) into $\prod_{l\in P} \eta_l$ problems, each of them having a linear function as an objective one. This linear function stems from the consideration of one of the linear functions that compound each link delay's approximation and summing them. Let us now use the index $j(l)$, $j(l)=1\ldots\eta_l$ for all $l\in P$, to denote the linear function chosen for link l, corresponding to subinterval $\Delta_{l,j}$. In order to consider each linear function only on the corresponding domain we introduce new constraints to the problem. That is, the solution has to be restricted to belong to the original polytope and at the same time to the set $\{t \in R^m | w_l^T t \in \Delta_{l,j}\}$. It can be readily proved that this set is equivalent to imposing restrictions on the load on each link. Thus it is itself a polytope. We represent the intersection of the original polytope and the new one, which is also a polytope, in matrix form as the set $\{t\in R^m | A^* t \leq b^*\}$, where the matrix A^* and the vector b^* define the intersection polytope.

Finally, in order to have a problem equivalent to problem (12) we consider all combinations of linear functions for each link, find the maximum over t for each combination and keep the combination which leads to the greatest value of the objective.

The mathematical formulation of the equivalent problem can be seen in (13), where the maximum on $j(1), j(2), \ldots, j(|P|)$ means that we consider the maximum obtained when we let each value $j(l)$ vary between 1 and η_l:

$$\max_{j(1),j(2),\ldots,j(|P|)} \begin{cases} \max\limits_{t} & \sum\limits_{l\in P} \alpha_{j(l)} w_l^T t + \beta_{j(l)} \\ \text{s.t.} & A^* t - b^* \leq 0 \end{cases} \tag{12}$$

DOI: 10.1002/nem

As we have claimed above, this method leads to a solution that is arbitrarily close to the exact solution of the original problem (2). The proof of this statement is shown in Appendix C.

Problem (13) can be solved computationally by performing a loop of $\prod_{l \in P} \eta_l$ iterations. Note that the problem solved on each iteration is a linear one, which is very easy to solve.

For obtaining the partition needed to define the piecewise linear function, we propose to iteratively compute the subintervals such that within each of them the maximum difference between the approximate function and the original one is a given n, at most. This constructive procedure is shown in Appendix C, and is part of the proof of achieving a solution arbitrarily close to the exact solution.

In order to reduce the number of iterations, we pre-compute the maximum value that the load can achieve at each link according to the constraints imposed by the polytope. Table 1 shows the number of subintervals needed to define the piecewise linear function, for different percentage errors and maximum link utilization (LU). This provides an idea of the complexity of the procedure. For example, for a six-link length path, at most $4^6 \approx 4000$ linear problems need to be resolved for obtaining aresult with 10% of error; thus the numerical complexity is still feasible.

5. NUMERICAL RESULTS

In order to assess the results of the numerical approximation method we shall first use the Abilene Network as before. We will also use the same polytope as before, so as to be able to compare results afterwards. As optimization software we use CPLEX [36]. We shall secondly perform further simulation studies on another network, in order to have more information about the computational time.

We compute the maximum delay for the four paths and three interdomain demands used in Section 3.2, using the numerical approximation method. The results, along with the exact solutions, are displayed in Figure 4, where the bars indicate the maximum error (10% in this case). Overall, the computation of each of the aforementioned values takes in mean 2.28 s. The maximum link utilization (imposed by the topology) was between 30% and 80%. These computational times are dramatically smaller than those necessary for obtaining the exact solution (approximately a $1000\times$ decrease), while providing a very tight bound.

The previous simulations allowed us to validate the method and show that its computational time is much smaller than that obtained through the exact method. We shall now explore this computational time when varying different parameters of the problem, namely the error, the maximum links' load and the number of links in the path.

The computational time depends on the accuracy needed by the application, which is not established a priori since it is a decision to be taken by each AS. It depends, in addition, on the maximum link utilizations allowed by the polytope along with the topology, and on the number of links compounding the path.

In order to asses the impact of the accuracy in the time consumed by the procedure we repeated the simulations allowing a maximum error of 5% and of 2%. The mean time needed for computing the maximum delay over one path for one interdomain demand is of 1 and 7.9 s, for obtaining a solution within 5% and 2% error respectively, which implies approximately a $90\times$ decrease for the 2% error, with respect to the exact method.

The previous results were obtained using historical traffic demands over the Abilene Network. This implies that the maximum link load, imposed by the polytope, is between 30% and 80%, as mentioned before. We now present further simulation results using synthetic data to define the polytope, so as to obtain results on a scenario with higher maximum link utilizations.

Table 1. Number of subintervals needed to define the piecewise linear function

	ε 1%	ε 10%
LU 60%	5	2
LU 80%	9	3
LU 90%	12	4

DOI: 10.1002/nem

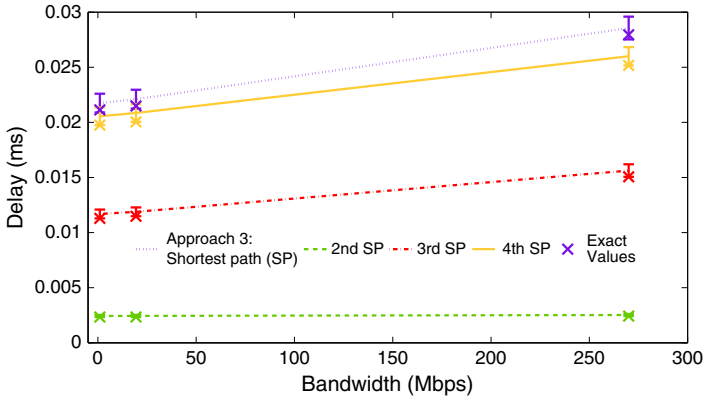


Figure 4. Approximate and exact solution: maximum delay for one origin–destination pair for different bandwidth demands and paths

Figure 5 shows the ratio of the time consumed for computing the delay bound through the approximate method to the time consumed by the exact method. The bound was computed for one interdomain demand of 19.4 Mbps for a polytope imposing maximum link utilizations between 80% and 90% and three different values of error (i.e. ε). This was repeated for the same four different paths presented above. Results of these simulations show that the time consumed by the approximate method is much less than that consumed by the exact method. In the worst case, that is, ε=2% and the fourth shortest path, the time consumed by the approximate method is 19 min, while the time consumed by the exact method is 38 min.

Finally, we shall explore the influence on the computational time of the number of links in the path. For doing so, let as utilize a larger network, the GÉANT network [37], which is composed of 23 nodes and 74 links. Figure 6 shows the computational time for one interdomain demand of 19.4 Mbps from an origin to a destination node, through different paths. Results are presented for a link utilization between 20% and 90% and different values of the allowed error. It is worth clarifying that the topology does not allow having high link utilizations in all links at the same time, since there are a number of bottleneck links on it. Results show that the computational time is not very sensitive to the path length. For the case of an 11-link path, which greatly exceeds the maximum path length on a domain, the time is approximately 7 min, providing a value within 2% error.

The approximate method was shown, through extensive simulation studies, to consume low computational times in most cases. The methods proposed in this work are conceived to be used by each AS in order to obtain a value of a metric to announce in the process of interdomain path selection. The announced bound is supposed to hold for a long period of time, e.g. several hours. In this context, the

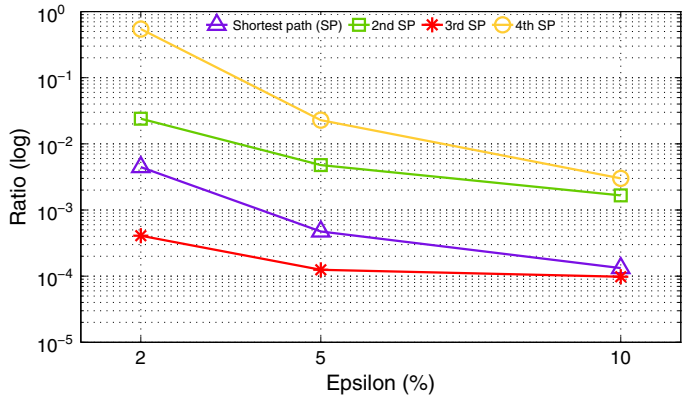


Figure 5. Ratio of the time consumed by the approximate method to the exact method: one interdomain demand for different allowed error and maximum link utilization between 80% and 90%. Bound computed for four different paths over the Abilene Network

DOI: 10.1002/nem

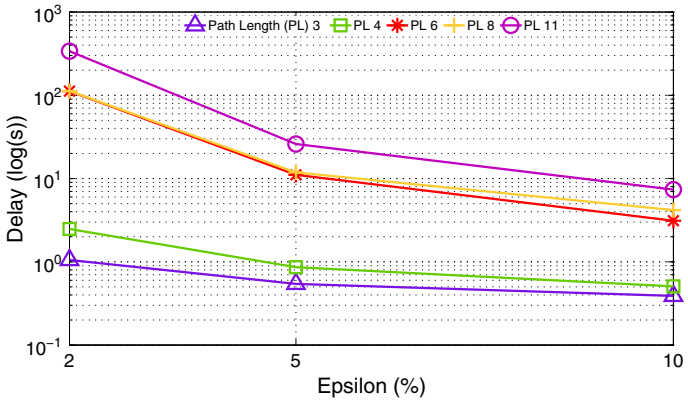


Figure 6. Time consumed by the approximate method: one interdomain demand for different allowed error and maximum link utilization between 20% and 90%. Bound computed for different paths over the GÉANT network

time consumed by the approximate method is considered totally acceptable. However, we have not focusedour work on optimizing this computational time, which could be, for instance, diminished through parallelizing the code, since its nature allows it (it solves an optimization problem over several independent feasible regions).

6. CONCLUSION AND FUTURE WORK

We have presented a means of computing a bound on the end-to-end delay. The method takes into account the uncertainties in the traffic traversing the AS, which have been modeled as a polytope. Therefore, it is a value that the AS can guarantee for a certain period of time. The problem was mathematically stated and different solutions were provided. A method for finding the exact solution was given, and an alternative approximate method was proposed, as a remedy for the high computational cost of the former. Such an approximate method renders, rather than a value of the delay, an interval to where the real value of the maximum delay is guaranteed to belong. The latter was theoretically proven and numerically validated, by comparing the results to the real maximum. Both methods were tested on real networks using measurement and synthetic data. The approximate method was shown, through simulations, to provide acceptable computational times on several scenarios. Altogether, we have proposed a method to enhance PCE-based interdomain path selection mechanisms, which can be implemented with low computation complexity and little monitoring infrastructure.

As future work, we shall explore the case of having uncertainty on the AS topology in addition to traffic uncertainty; for instance, taking into account the case of link or node failures, and being able to provide even in those cases a tight end-to-end delay bound. We shall also explore the possibility of building a delay curve as a function of ingress traffic. If the delay can be advertised as a function of ingress traffic, this would allow more sophisticated routing decisions to be made, leading to lower end-to-end delay values. In addition, in the scenario considered in this paper where ASs collaborate among them, the problem of distributing all flows among the ASs involved for achieving social welfare at the time they have QoS requirements becomes an interesting one and raises new questions. Studying such scenario using the present approach as a building block is a challenge that we shall address in the future.

APPENDIX A: CONVEXITY OF THE OBJECTIVE FUNCTION

In this section we prove that the objective function of problem (2) is a convex function but not a strictly convex one. We use the notation introduced in Section 2.1. In addition, we will denote the objective function of problem (2) as $f(t)$ and the feasible region of such a problem as S.

We are interested in finding out if $f(t)$ is a convex function over S; thus we shall explore whether the following inequality holds [38]:

$$f(t_1) \geq f(t_2) + \nabla f(t_2)^T (t_1 - t_2), t_1, t_2 \in S \tag{13}$$

Applying the definition of f to equation (13) we obtain the following inequality:

$$\sum_{l \in P} \frac{1/c_l}{1 - w_l^T t_1 - u/c_l} \geq \sum_{l \in P} \frac{1/c_l}{1 - w_l^T t_2 - u/c_l} + \sum_{l \in P} \frac{1/c_l w_l^T (t_1 - t_2)}{\left(1 - w_l^T t_2 - u/c_l\right)^2}, t_1, t_2 \in S \tag{14}$$

Let us now define $g_l(t)$, an auxiliary function, in order to simplify the notation, as

$$g_l(t) = 1 - w_l^T t - u/c_l, t \in S \tag{15}$$

Substituting the latter definition in equation (14) and performing some regular math operations, we obtain the following inequality:

$$\sum_{l \in P} \frac{\left(g_l(t_2) - g_l(t_1)\right)^2}{g_l(t_1) g_l(t_2)^2} \geq 0, t_1, t_2 \in S \tag{16}$$

Note that each term in inequality (16) is either zero or greater than zero for all $t_1, t_2 \in S$. Therefore, the function f is convex over S. It remains to be shown whether the function is strictly convex or not.

Therefore, we need to find out if there exist t_1 and $t_2 \in S$ such that $g_l(t_2) - g_l(t_1)$ is equal to zero for all $l \in P$. The latter is equivalent to showing that there exist t_1 and $t_2 \in S$ such that $< w_l, t_2 - t_1 >$, the scalar product of vector w_l and $(t_2 - t_1)$, is equal to zero for all $l \in P$; that is, having all vectors w_l, $l \in P$ orthogonal to the vector $(t_2 - t_1)$. Since the vectors w_l do not form a basis of R^m it is possible to find t_1 and $t_2 \in S$ such that their difference is orthogonal to all vectors w_l, $l \in P$. This completes the proof, which concludes that f is a convex function, but not a strictly convex one.

APPENDIX B: CONVEXITY OF $J(x)$ AND CHARACTERIZATION OF THE OPTIMAL POINT

In this appendix we shall prove that the function $J(x)$ defined in Section 3 is indeed a strictly convex function over V_x and that the solution to Problem (10) is an extreme point of the polytope V_x. We shall use the notation introduced in Sections 2.1 and 3.

Let us define $\lambda_l(x)$ as

$$\lambda_l(x) = \left(1 - v_l^T x - u/c_l\right)^{-2}, \forall l \in P \tag{17}$$

and the matrix Λ as

$$\Lambda(x) = \operatorname{diag}\left(\lambda_1, \ldots, \lambda_{|P|}\right) \tag{18}$$

For all $x \in V_x$ and $l \in \{1 \ldots |P|\}$, $\lambda_l(x) > 0$. Thus $\Lambda(x)$ is a positive-definite matrix.[1]

In addition, we can check that $\left[v_1 \ldots v_{|P|}\right] = W_1^T W_1$ is also a positive-definite matrix. Thus, the Hessian of $J(x)$, which is

$$\nabla^2 J(x) = \left(W_1^T W_1\right) \Lambda(x) \left(W_1^T W_1\right) \tag{19}$$

is as well a positive-definite matrix, which concludes the proof.

We are now able to show that the solution to problem (10) is attained at an extreme point of V_x.

Indeed, we can prove by contradiction that the maximum of $J(x)$ over V_x must be reached at an extreme point of V_x. Since J is a strictly convex function, inequality (20) holds [38]:

[1] A $n \times n$ real symmetric matrix M is positive-definite if $z^T M z > 0$ for all non-zero vectors z, $z \in R^n$.

DOI: 10.1002/nem

$$J(\Phi) > J(\theta) + \nabla J(\theta)^T(\Phi - \theta), \forall \theta, \Phi \in V_x \tag{20}$$

Now, let $\bar{\theta} \in V_x$ be an optimal point of problem (7). Therefore, $\bar{\theta}$ is a strict maximum, since J is strictly convex, and, for all $\Phi \in V_x \backslash \left\{\bar{\theta}\right\}$, we must have

$$J(\Phi) - J\left(\bar{\theta}\right) < 0 \tag{21}$$

Together with inequality (20), we get

$$\nabla J\left(\bar{\theta}\right)^T\left(\Phi - \bar{\theta}\right) < 0, \forall \Phi \in V_x \backslash \left\{\bar{\theta}\right\} \tag{22}$$

By contradiction we suppose that $\bar{\theta}$ is not an extreme point of V_x. Then there exists $\mu \in R^{|P|}$ such that $\|\mu\| > 0$ and $\bar{\theta} + \mu, \bar{\theta} - \mu \in V_x$. By letting $\Phi = \bar{\theta} - \mu$ and $\Phi = \bar{\theta} - \mu$ at a time, we would get

$$\nabla J\left(\bar{\theta}\right)^T \mu < 0 \text{ and } -\nabla J\left(\bar{\theta}\right)^T \mu < 0 \tag{23}$$

which is not possible.

This allows us to conclude that $\bar{\theta}$ cannot be a non-extremal point of V_x, which finalizes the proof.

APPENDIX C: VALIDATION OF THE APPROXIMATE METHOD

In this section we prove that the approximate method presented in Section 4 reaches the exact solution of problem (2), up to some controlled error.

We remind the reader that we start from a problem like (2). We then partition the feasible region into several sub-regions and approximate the objective function of problem (2) as a linear function within each of the sub-regions. This problem can be seen in problem (13). We then keep the maximum of the output of all the possible problems of maximization over t in problem (13).

Let us refer to the original function, defined in problem (2), as *f(t)* and the piecewise linear approximation of f as $\tilde{f}$. Let $\tilde{t}_f$ and t_f be the values at which the maximum of $\tilde{f}$ and f are attained respectively. Let us also note the feasible region of problem (2) as S.

We set the hypothesis that for a given real positive ε, the approximation of f can be made such that the difference between f and $\tilde{f}$ is bounded by ε, i.e.

$$f(t) - \tilde{f}(t) \leq \varepsilon \, \forall t \in S \tag{24}$$

Under the conditions of equation (24) and with the definitions of $\tilde{t}_f$ and t_f provided above, we can prove that

$$f\left(t_f\right) - f\left(\tilde{t}_f\right) \leq \varepsilon \tag{25}$$

Let us first prove that equation (24) holds for the case of the M/M/1 model mean delay function. Note that for other functions this is a hypothesis to be checked before applying the algorithm. We provide a constructive proof shown in the following. For the sake of simplicity on the notation we will not include u in the formulation, but the whole procedure can be reproduced in an analogous way considering u.

Let $e(y) = \frac{1}{1-y}$ be such that $f(t) = \sum_{l \in P} 1/c_l e\left(w_l^T t\right), t \in S$. Let us note the partition of the domain of e over y ($[0,1)$) as the set of subsets Δ_i where

$$\Delta = \{\Delta_i : i = 1 \ldots \eta\} \tag{26}$$

Let $\tilde{e}$ be the piecewise linear approximation of e over each of the subsets defined in (26), such that the following inequality holds:

$$|e(y) - \tilde{e}(y)| \leq \frac{\varepsilon}{\sum_{l \in P} 1/c_l} = \delta \ \forall y \in \Delta \tag{27}$$

This will ensure that equation (24) holds since $\tilde{f}(t) = \sum_{l \in P} 1/c_l \tilde{e}\left(w_l^T t\right)$.

We shall consider the graphic displayed in Figure 7. Let us note Δ_i as $\Delta_i=[y_{i-1},y_i]$, with $y_0=0$. We define $z_i \in \Delta_i$ as the linearization point of function e in Δ_i. Let us define n_{i-1} and ε_i as the difference between e and $\tilde{e}$ at each y_{i-1} and y_i respectively, i.e.

$$\varepsilon_{i-1} = e(y_{i-1}) - \tilde{e}(y_{i-1}) \ \text{ and } \ \varepsilon_i = e(y_i) - \tilde{e}(y_i), i = 1 \ldots \eta \tag{28}$$

It is not difficult to see that the maximum of the difference between e and $\tilde{e}$ will be attained at either y_{i-1} or y_i, $i \in 1 \ldots \eta$.

Given y_{i-1} fixed, let z_i increase from y_{i-1}. We define y_i such that $\varepsilon_i=\delta$. As z_i increases, y_i and ε_{i-1} increase. Therefore, the maximum subinterval size under the constraint $\varepsilon_{i-1}, \varepsilon_i \leq \delta$ is achieved when

$$\varepsilon_{i-1} = \varepsilon_i = \delta \tag{29}$$

Therefore, given y_{i-1} and δ, we can find a value z_i imposing that

$$e(y_{i-1}) - \tilde{e}(y_{i-1}) = \delta \tag{30}$$

Once z_i is known, we can compute y_i by imposing

$$e(y_i) - \tilde{e}(y_i) = \delta \tag{31}$$

All in all, it appears that e can be approximated over its domain by means of a piecewise linear function. Thus $f(t) = \sum_{l \in P} 1/c_l e\left(w_l^T t\right)$ can be approximated by means of a sum of piecewise linear functions, and this approximation, which we note as $\tilde{f}(t)$, is such that equation (24) holds for all $t \in S$.

We have provided a constructive proof of equation (24). We are now able to show that (25) holds. Indeed, let again $\tilde{t}_f$ and t_f be the values at which the maximum of $\tilde{f}$ and f are attained respectively. The following inequalities are obtained straightforwardly from the definition of the maximum

$$\tilde{f}\left(\tilde{t}_f\right) \geq \tilde{f}\left(t_f\right) \tag{32}$$

$$f\left(t_f\right) \geq f\left(\tilde{t}_f\right) \tag{33}$$

We are interested in finding a bound to the difference $f(t_f)-f(\tilde{t}_f)$, which can be rewritten as in the following equation:

$$f\left(t_f\right) - f\left(\tilde{t}_f\right) = \overbrace{\tilde{f}\left(\tilde{t}_f\right) - f\left(\tilde{t}_f\right)}^{\leq 0} + \overbrace{\tilde{f}\left(t_f\right) - \tilde{f}\left(\tilde{t}_f\right)}^{\leq 0} + \overbrace{f\left(t_f\right) - \tilde{f}\left(t_f\right)}^{\leq \varepsilon} \tag{34}$$

Equation (34) immediately leads to the inequality $f(t_f)-f(\tilde{t}_f) \leq \varepsilon$, which completes the proof. Note

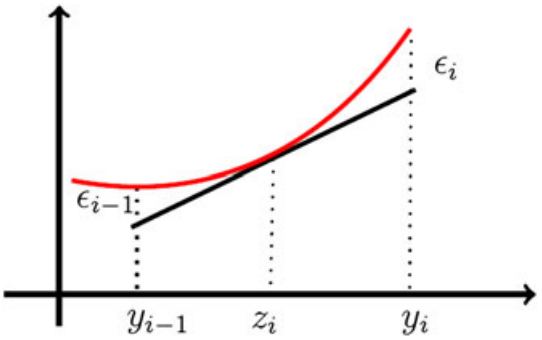


Figure 7. Difference between function e and its approximation $\tilde{e}$

that in equation (34) we have used the fact that $f(t)$ is greater than $\tilde{f}(t)$ for all $t \in S$, which is true since $\tilde{f}$ is the piecewise linear approximation of a convex function.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreement 248567 for the ETICS Project. Further information is available at www.ict-etics.eu. This work was also partially funded by the Uruguayan Agency for Research and Innovation (ANII) under grant PR-POS-2008-003.

REFERENCES

1. Zhang R, Vasseur JP. Traffic engineering (TE) requirements. *RFC 4216: MPLS Inter-Autonomous System (AS)*. Internet Engineering Task Force, November 2005.
2. Farrel A, Vasseur JP, Ayyangar A. *RFC 4726: A Framework for Inter-Domain Multiprotocol Label Switch*. Internet Engineering Task Force, 2006.
3. Awduche D, Berger L, Gan D, Li T, Srinivasan V, Swallow G. *RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels*. Internet Engineering Task Force, December 2001.
4. Path computation element (PCE) IETF working group. Available: http://www.ietf.org/html.charters/pce-charter.html [June 2011].
5. King D, Farrel A. The application of the path computation element architecture to the determination of a sequence of domains in MPLS and GMPLS July 2009.
6. Pouyllau H, Douville R, Djarallah NB, Le Sauze N. Economic and technical propositions for inter-domain services. *Bell Labs Technical Journal* 2009; **14**: 185–202.
7. Bertrand G, Texier G. Ad-hoc recursive PCE based inter-domain path computation (ARPC) methods. In *Fifth International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs)*, 2008.
8. Pelsser C, Bonaventure O. Path selection techniques to establish constrained interdomain MPLS LSPs. In *Networking 2006*. LNCS Vol. 3976, Springer: Berlin, 2006; 209–220.
9. Quoitin B, Pelsser C, Swinnen L, Bonaventure O, Uhlig S. Interdomain traffic engineering with BGP. *Communications Magazine, IEEE* 2003; **41**: 122–128.
10. Yannuzzi M, Fonte A, Masip-Bruin X, Monteiro E, Sanchez-Lopez S, Domingo-Pascual J. A self-adaptive QoS routing framework for multihomed stub autonomous systems. In *Proceedings of Eunice, IFIP*, Madrid, Spain, 2004; 241–247.
11. Lima SR, Carvalho P, Freitas V. Toward scalable management of multiple service levels in IP networks. In *Proceedings of the Fourth European Conference on Universal Multiservice Networks*, IEEE Computer Society, Washington, DC,2007; 223–232.
12. Niccolini S, Molina M, Raspall F, Tartarelli S. Design and implementation of a one way delay passive measurement system. In *Network Operations and Management Symposium, 2004*. IEEE/IFIP, Vol. 1, 2004; 469–482.
13. Hernandez A, Magana E. One-way delay measurement and characterization. In *Third International Conference on Networking and Services, 2007*. ICNS, 2007; 114.
14. Corral J, Texier G, Toutain L. End-to-end active measurement architecture in IP Networks (SATURNE). In *PAM 2003: Workshop on Passive and Active Measurements*, 2003; 3–4.
15. Shalunov S, Teitelbaum B, Karp A, Boote J, Zekauskas M. *RFC 4656: A One-way Active Measurement Protocol (OWAMP)*. Internet Engineering Task Force, September 2006.
16. De Vito L, Rapuano S, Tomaciello L. One-way delay measurement: state of the art. In *IEEE Transactions on Instrumentation and Measurement* 2008; **57**: 2742–2750.
17. Masip-Bruin X, Sanchez-Lopez S, Sole-Pareta J, Domingo-Pascual J. QoS routing algorithms under inaccurate routing for bandwidth constrained applications. In *ICC '03*, IEEE, 2003; 1743–1748.
18. Guérin RA, Orda A. QoS routing in networks with inaccurate information: theory and algorithms. *IEEE/ACM Transactions on Networking* 1999; **7**: 350–364.
19. Levendovszky J, Orosz C. Developing novel statistical bandwidths for communication networks with incomplete information. In *Symposium on Experimental and Efficient Algorithms*, 2005; 614–617.
20. Claise B. *RFC 3954: Cisco Systems NetFlow Services Export Version 9*. Internet Engineering Task Force, 2004.
21. Vardi Y. Network tomography: estimating source–destination traffic intensities from link data. *Journal of the American Statistical Association* 1996; **91**(433): 365–377.
22. Zhang Y, Roughan M, Lund C, Donoho DL. Estimating point-to-point and point-to-multipoint traffic matrices: an information-theoretic approach. *IEEE/ACM Transactions on Networking* 2005; **13**: 947–960.
23. Casas P, Vaton S, Fillatre L, Chonavel T. Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks. In *21st International Teletraffic Congress*, 2009; 1–8.
24. Teixeira R, Duffield N, Rexford J, Roughan M. Traffic matrix reloaded: impact of routing changes. In *Passive and Active Network Measurement*. LNCS Vol. 3431, Springer: Berlin, 2005; 251–264.

25. Casas P, Vaton S, Fillatre L, Nikiforov I. Optimal volume anomaly detection and isolation in large-scale IP networks using coarse-grained measurements. *Computer Networks* 2010; **54**: 1750–1766.
26. Ben-Ameur W, Kerivin H. Routing of uncertain traffic demands. *Optimization and Engineering* 2005; **6**: 283–313.
27. Larroca F, Rougier JL. Robust regression for minimum-delay load-balancing. In *21st International Teletraffic Congress*, 2009; 1–8.
28. Duffield NG, Goyal P, Greenberg A, Mishra P, Ramakrishnan KK, Van der Merive JE. A flexible model for resource management in virtual private networks. In *SIGCOMM '99*. ACM: New York, 1999; 95–108.
29. Johansson M, Gunnar A. Data-driven traffic engineering: techniques, experiences and challenges. In: *3rd International Conference on Broadband Communications, Networks and Systems*, 2006; 1–10.
30. Fillatre L, Marakov D, Vaton S. Forecasting seasonal traffic flows. In *Workshop on QoS and Traffic Control*, 2005.
31. Brondsted A. *An Introduction to Convex Polytopes*. Springer: Berlin, 1982.
32. Jones CN, Polyhedral tools for control. PhD thesis, University of Cambridge, UK (2005).
33. Zhang Y. The Abilene Dataset. Available: http://www.cs.utexas.edu/ yzhang/research/AbileneTM/ [June 2011].
34. Multi-Parametric Toolbox (MPT). A tool (not only) for multi-parametric optimization. Available: http://control.ee.ethz.ch/mpt [June 2011].
35. Khachiyan L, Boros E, Borys K, Elbassioni K, Gurvich V. Generating all vertices of a polyhedron is hard. In *Proceedings of the 17th annual ACM-SIAM symposium on Discrete algorithm, SODA '06, ACM*, New York, NY, USA, 2006; 758–765.
36. IBM ILOG CPLEX Optimizer. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/ [June 2011].
37. The GÉANT network. Available: http://www.geant.net [June 2011].
38. Minoux M. Mathematical Programming: Theory and Algorithms. Wiley: Chichester, 1986.

AUTHORS' BIOGRAPHIES

Isabel Amigo is a PhD student at Telecom Bretagne (France) and at Universidad de la República (Uruguay). Her research topics are related to Quality of Service in interdomain networks, and are performed in the context of the project ETICS, a large scale integrating project funded by the European Commission. Since 2007 she holds a degree in Electrical Engineering (UdelaR), her final thesis was related to Traffic Engineering in multiservice networks. Since 2006 she works in the Electrical Engineering department (IIE) of Universidad de la República as an Assistant. From the end of 2007 until beginning 2010 she gained professional experience at a Uruguayan National project.

Sandrine Vaton is an associate professor at Telecom Bretagne, Brest, France. She obtained the engineering degree from Telecom ParisTech in 1994, a Master of Science degree in Probabilities from the university of Paris 6 in 1995 and a PhD in statistical signal processing from Telecom ParisTech in 1998. Since 1999 she is an associate professor at Telecom Bretagne. She is in charge of lectures in performance evaluation, statistical methods and cryptography. Her main interest in research concern traffic analysis from a statistical point of view for applications to Quality of Service and security of telecommunication networks. Since 1999 she has been the advisor of five PhD students, several post-doc students and many internship students. She participated to the networks of excellence Euro-NGI and Euro-NF, to the OSCAR project funded by the National Research Agency, to several collaborative research projects funded by the Institut Telecom. and to ECOS-Sud and STIM AmSud projects with South America. She is also involved in bilateral research collaborations with the industry. Ongoing research projects are the VIPEER project funded by the french national research agency, and the ETICS and DEMONS projects which are two large scale integrating projects funded by the European Commission.

Thierry Chonavel has been working at Telecom Bretagne since 1995, where he has been full professor since 2003. He is mainly involved in signal processing teaching and research activities. His interest ranges from array processing (covariance matrix estimation, adaptive eigenvalue decomposition), underwater acoustics (multi-paths channel estimation by means of penalized and bayesian techniques) and acoustics seismic (blind deconvolution of seismic records) spread-spectrum communications (iterative and turbo receivers design, multiple output radiocommunication platform), speech processing (voice modelling and voice conversion), GPS-Galileo receivers (multi-paths deconvolution and interference mitigation of radio-astronomy observations), radar processing (waveforms design to achieve good ambiguity functions and estimators performance bounds) for airborne systems and driving aid, speech compression (by means of empirical mode decompositions and amplitude-frequency decompositions), MIMO techniques for powerline communications and nonlinear state space models. His contributions in Networking are related to works involving Kalman filtering and optimization.

Federico Larroca is an Assistant Professor at the Engineering School of the Universidad de la República (Uruguay). He was a research engineering (PostDoc) at Telecom ParisTech (ex ENST) during the first quarter of 2010, where he obtained his Ph.D. degree in Computer Science and Networking under the advisoring of Prof. Jean-Louis Rougier in December 2009. He received the degree in Telecommunication Engineering in 2006 from the Universidad de la República, with a thesis concerning the development of a propagation model and loss prediction software for cellular signals in suburban and urban environments. From 2004 to 2011 he held a teaching assistant position at the Universidad de la República. His current research interests are related to statistical characterization and analysis of network traffic, traffic engineering, and performance analysis in heterogeneous networks.

Int. J. Network Mgmt 2012; **22**: 162–179
DOI: 10.1002/nem

CLASSIFICATION DE TRAFIC

CONTEXTE ET ETAT DE L'ART

La classification de trafic consiste à associer le trafic réseau avec l'application ou la catégorie d'applications qui l'a généré. Les opérateurs aimeraient dans certains cas savoir à quelle application appartiennent les paquets de façon à optimiser l'ingénierie du trafic, à affiner leur politique de tarification, etc... Tous les opérateurs surveillent en permanence la composition du trafic par catégorie d'applications. Ils analysent les tendances d'usage et aimeraient anticiper l'émergence de nouvelles applications consommatrices en bande passante. Les solutions de Qualité de Service (QoS) qui séparent le trafic en classes et leur affectent différentes priorités nécessitent également la capacité d'associer le trafic aux applications puisque celles-ci n'étiquètent pas elles-mêmes le trafic qu'elles génèrent. La classification de trafic peut également être utile pour faire de la tarification différenciée ou pour vérifier les paramètres techniques de certains contrats. Les lois sur l'interception du trafic illégal obligent également les fournisseurs de service à analyser le trafic de leurs clients et à reconnaître le trafic illégal ou suspect.

La classification de trafic est devenue difficile pour différentes raisons. Tout d'abord, les opérateurs ont à analyser une quantité considérable de trafic. Le rapport annuel CISCO [20] estimait le trafic Internet total à 20 exabytes par mois en 2010 et prévoyait un trafic de 1 zettabyte en 2015. Les volumes et les débits de trafic à analyser plaident pour le développement de méthodes légères pour la classification du trafic. D'autre part, les techniques traditionnelles pour la classification de trafic ont des limitations. Traditionellement la classification pouvait se baser soit sur l'analyse du numéro de port, soit sur la reconnaissance de chaînes de caractères spécifiques dans la charge utile du paquet (DPI, *Deep Packet Inspection*). La classification basée sur le numéro de port n'est plus fiable car beaucoup d'applications changent dynamiquement leur numéro de port ou se cachent derrière d'autres applications en utilisant leur numéro de port (par exemple, port 80 pour http). Les techniques de DPI reconnaissent des chaînes de caractères spécifiques dans la charge utile. Certaines applications peuvent échapper à cette analyse en chiffrant leur trafic. De plus ces techniques deviennent extrêmement lourdes quand le nombre de règles de filtrage et le volume de trafic à analyser augmentent.

Il était donc nécessaire de développer de méthodes qui ne se basent pas ou pas uniquement sur le numéro de port ou l'analyse de la charge utile. La littérature dans le domaine de la classification de trafic est abondante. Cet axe de recherche a été développé par différentes équipes académiques à partir de 2005 [46] [37] [48] [42] [9] [29] [26] [5] [25] [35] [31] [44] [52] [8]. L'abondance de la littérature démontre l'intérêt de la communauté pour ce sujet. Il existe également quelques états de l'art sur le sujet, comme par exemple [43], [50], [57].

Les techniques de reconnaissance applicative basées sur de l'apprentissage automatique sont classiquement divisées en catégories en fonction des descripteurs de trafic qu'elles considèrent. Une taxonomie habituelle est de considérer deux catégories : les classificateurs statistiques et les classificateurs comportementaux. Les classificateurs statistiques associent une application à un flot, c'est-à-dire à une suite de paquets partageant le même 5-uplet formé des deux adresses IP (source, destination), des deux numéros de port de couche transport et du type de protocole de transport. Elles basent leur décision sur des descripteurs de trafic qui ne nécessitent pas d'analyser la charge utile des paquets. Les descipteurs habituels sont, par exemple, des tailles ou instants de passage de paquets, certains champs de l'entête de couche transport comme les drapeaux TCP, ou d'autres propriétés des flots comme ceux qui sont générés sous forme de rapport par les sondes NetFlow (standardisé à l'IETF par le groupe IPFIX). Les méthodes comportementales [8] analysent le comportement particulier d'un hôte et en particulier les échanges de cet hôte avec d'autres machines, différents types de serveurs, de clients, ou des pairs si il s'agit de trafic pair à pair (P2P, *Peer to Peer*).

Les techniques d'apprentissage automatique sont divisées en deux catégories : les méthodes supervisées et les méthodes non supervisées. Les méthodes supervisées nécessitent l'apprentissage d'un modèle sur un jeu de données étiqueté avec une *groundtruth*. Dans le cas du problème de reconnaissance applicative c'est une information fiable sur l'application ayant généré un flot qui constitue la *groundtruth*. Les méthodes permettant d'établir cette *grountruth* peuvent être soit une DPI en utilisant un outil comme L7-Filter [21] soit une classification par numéro de port, pour les traces antérieures à 2002 pour lesquelles cette classification était encore fiable, en utilisant un outil comme CoralReef [11] de CAIDA. Dans le cas où les mesures à étiqueter ne sont pas un trafic de charge, mesuré passivement en un point d'un réseau, mais un trafic généré en laboratoire dans un environnement contrôlé pour les besoins de la métrologie la *groundtruth* peut être obtenue grâce à l'outil GT [35]. Cet outil, installé sur une machine cliente, analyse les appels systèmes générés par les différentes applications et les corrèle avec le trafic mesuré au niveau d'une interface réseau ; il effectue également une DPI du trafic capturé.

Il ne peut pas y avoir de consensus sur une définition universelle de la *groundtruth* ; celle-ci dépend beaucoup de la façon dont elle a été obtenue. Si on prend l'exemple simple d'une classification par DPI les étiquettes dépendront des filtres sur lesquels se base la reconnaissance d'expressions régulières dans la charge utile des paquets. D'une manière générale l'apprentissage des modèles doit être fait avec beaucoup de soin car il biaise les résultats de la classification. La stabilité des modèles dans le temps et dans l'espace est également un problème. Bien que certaines études [44][31] soient rassurantes sur ce point, l'évolution dans les usages de l'Internet, les mises à jour du code des applications, sont autant de facteurs qui plaident pour une mise à jour régulière des modèles de trafic associés aux applications l'ayant généré. De la même façon, si certains descripteurs de trafic comme les tailles de paquets peuvent être stables lorsque les captures se font dans différents réseaux ou en différents points d'un réseau, d'autres descripteurs comme le temps inter-paquets peuvent être impactés par le réseau [40].

L'obtention de traces étiquetées est compliquée. Les opérateurs ne souhaitent pas d'une manière générale donner accès aux mesures qu'ils font sur leur réseau. Ils le font parfois en protégeant leurs données par un accord de confidentialité (NDA, *Non Disclosure Agreement*). Le problème du respect de la vie privée des usagers peut être résolu par l'anonymisation des traces. Un autre problème de confidentialité important vient des accords entre opérateurs sur les échanges de trafic (accords client/fournisseur ou accords de peering). Du fait de ces accords et de la compétition entre opérateurs ces derniers souhaitent révéler le moins d'information possible sur la composition du trafic échangé [27]. La communauté de chercheurs a fait des efforts pour produire un ensemble de traces publiques qui peuvent servir pour des analyses [10] [56] [55] mais il s'agit bien souvent de mesures obtenues sur des réseaux de campus. Il serait intéressant d'avoir une plus grande diversité dans le type de mesures car la composition du trafic diffère sensiblement entre un réseau résidentiel et un réseau d'entreprise, entre un réseau d'accès et un coeur de réseau, ou en fonction des différentes zones géographiques dans le monde et le scénario considéré a un impact sur les résultats de l'analyse.

Les performances des différentes techniques de classification de trafic ont été largement étudiées depuis 2005 en ce qui concerne les taux de classification (i.e. % de flots qui sont correctement associés à l'application qui les a générés). Mais en dépit d'une littérature pléthorique sur le sujet la question de l'implémentation *pratique* de ces méthodes pour l'analyse hors ligne de gros volumes de trafic ou l'analyse en temps réel et à débit élevé n'a pas reçu assez d'attention. Il existe quelques études qui s'intéressent à l'impact sur les taux de classification de techniques permettant une montée en débit, par exemple le sous-échantillonage au niveau flot [51] ou la sélection d'un petit nombre de descripteurs de trafic suffisamment informatifs pour permettre une classification précise [58]. Mais la littérature sur l'accélération des techniques légères de reconnaissance applicative est quasiment inexistante, que l'on considère une approche logicielle (parallélisation de code sur des architectures multi-coeur, grappes de calcul, calcul scientifique sur carte graphique ...) ou une approche matérielle (conception de circuits dédiés en ASIC ou en FPGA).

A titre de comparaison on peut considérer le problème de la classification de paquets par les équipements de réseau en fonction des résultats d'une recherche sur un ensemble de filtres utilisant différents champs de l'entête comme clé (DPI). Dans ce domaine il y a clairement eu deux axes de recherche, l'un algorithmique et l'autre architectural [54] [7]. Au contraire, bien que certaines études mentionnent que les techniques d'apprentissage automatique (*machine learning*) ne seraient pas capables de monter au débit des interfaces pour permettre une reconnaissance applicative en temps réel, le domaine de la conception pratique de ces systèmes n'a pas été suffisamment exploré, en dehors de quelques contributions récentes

[45] [41] [12] [30] [34].

Le domaine de la classification de trafic a été très actif depuis les premiers travaux en 2004-2005. Depuis 2010 j'organise un workshop annuel avec Christian Callegari de l'université de Pise : TRAC, International Workshop on TRaffic Analysis and Classification [17][18] [19]. TRAC est co-localisé avec la conférence IEEE IWCMC. Le workshop s'intéresse à toutes les études relatives à la surveillance et à l'analyse du trafic même si la classification de trafic en est un thème privilégié, plusieurs sessions lui ayant été dédiées en 2010 et 2011. TRAC permet aux chercheurs du domaine de se rencontrer et de faire un bilan annuel des recherches en cours.

D'autres workshops ou conférences mettent la surveillance du trafic au coeur de leur thématiques. On peut citer en particulier les conférences IMC [1] (ACM SigComm Internet Measurement Conference), PAM [2] (Passive and Active Measurement Conference) et le workshop COST TMA [24] (COST Traffic Monitoring and Analysis Workshop) sans que cette liste ne soit exhaustive.

Dans la suite de ce chapitre nous présentons deux contributions dans le domaine de la reconnaissance de trafic par des méthodes d'apprentissage automatique.

DEUX CONTRIBUTIONS

DAHMOUNI Hamza, VATON Sandrine, ROSSÉ David, A Markovian signature-based approach to IP traffic classification, ACM SIGMETRICS 2007, Mining Network Data Workshop (MineNet), June 12, San Diego, CA , 2007, pp. 29-34

Résumé de l'article ACM SIGMETRICS MineNet 2007

Dans l'article ACM Sigmetrics MineNet 2007 [26] nous proposons un classificateur statistique basé sur l'analyse des drapeaux TCP des paquets successifs d'un flot. Le champ *TCP Flags* est un champ de 6 bits {URG, ACK, PSH, RST, SYN, FIN} dans l'entête du protocole de transport de couche 4 TCP. La suite de drapeaux TCP est modélisée comme une Chaîne de Markov à Temps Discret (CMTD) et à états finis. Cette CMTD est caractérisée par une distribution initiale et une matrice des probabilités de transition entre états. L'apprentissage des paramètres de cette CMTD se fait sur une trace étiquetée après une reconstruction des flots. Les valeurs estimées des probabilités initiales et des probabilités de transition se basent sur un comptage du nombre d'événements correspondant, par exemple, à une transition sur deux paquets successifs d'un même flot. L'article propose différents critères pour la classification de nouveaux flots : un critère du Maximum de Vraissemblance (ML, Maximum Likelihood) et un critère de Neyman-Pearson (NP, Neyman-Pearson). La règle ML est optimale asymptotiquement c'est-à-dire quand le nombre de paquets pris en compte pour la décision tend vers l'infini. La règle NP s'applique au cas d'une décision entre deux applications : cette règle est optimale au sens où le test NP est le test le plus puissant, c'est-à-dire de probabilité de détection maximale parmi tous les tests dont la probabilité de fausse alarme est bornée supérieurement par une valeur α. Les performances de la méthode sont ensuite testées sur du trafic Orange France ; la campagne de mesure datait de 2006-2007 sur un réseau GPRS Orange et le point de capture se situait au niveau de l'interface Gb (entre SGSN et PCU). La méthode a démontré sa capacité à discriminer du trafic HTTP et Telnet sur la base des 6 à 10 premiers paquets d'un flot monodirectionnel.

Plateforme de mesure et classification de trafic à Télécom Bretagne

Malgré ces premiers résultats encourageants de nombreuses questions se posaient. La validation sur les données GPRS Orange considère un cas extrêmement favorable, puisqu'il s'agit de prendre une décision entre deux applications seulement, HTTP et Telnet dans ce cas. Les opérateurs ont en réalité affaire à une très grande diversité d'applications. Cette tendance est particulièrement accentuée par l'engouement pour les smartphones et les applications mobiles [53]. Des études évaluent à plus de 50 000 le nombre d'applications pour iPhone, la plupart d'entre elles ayant des fonctionnalités réseau [22]. Il est certes possible de généraliser la méthode proposée au cas de plusieurs applications, par exemple en prenant des décisions "un contre un" puis en concluant en fonction du vote majoritaire. Par ailleurs le critère ML ne se limite pas, contrairement au critère NP, au cas de deux applications et la plupart du temps, on ne cherche

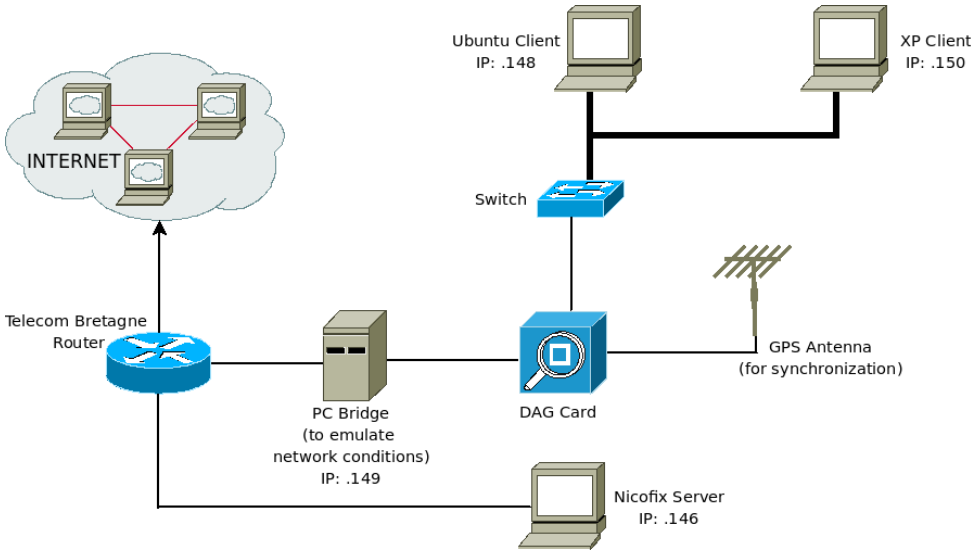


FIGURE 1 – Plateforme de métrologie à Télécom Bretagne

pas a reconnaître des applications mais des catégories d'applications, beaucoup moins nombreuses.

Une plateforme a été mise en place à Télécom Bretagne dans le cadre de projets d'étudiants pour des expérimentations sur la mesure de trafic (voir Figure 1). L'acquisition est faite par une carte DAG [28] que nous avions acquise pour les besoins du projet ANR OSCAR [3]. Ces cartes de la société Endace sont spécialisées dans la capture de traces de niveau paquet au débit maximum des interfaces, sans perte, et avec un horodatage précis des paquets obtenu grâce à une antenne GPS et un boîtier de synchronisation Acutime de la société Trimble. La carte DAG, hébergée sur une machine hôte, est en coupure entre un certain nombre de machines clientes et l'accès Internet. La carte mesure donc le trafic bidirectionnel généré par les machines clientes. Une machine supplémentaire a été ajoutée entre l'accès Internet et la carte DAG. Cette machine configurée comme un *bridge* sert à ajouter artificiellement différentes perturbations (pertes, délai, gigue, etc...) avec l'outil de contrôle de trafic TC [47] pour émuler la variabilité des conditions réseau et, par exemple, en mesurer l'impact sur les performances de la classification.

Une application a été développée par des étudiants en stage ou en projet. Un agent logiciel reconstruit les flots à partir des mesures brutes au niveau paquet, et construit pour chaque flot un rapport incluant le 5-uplet ainsi que les drapeaux TCP des N premiers paquets du flot. Un autre agent est chargé de l'apprentissage du modèle de chaîne de Markov caractérisant la succession des drapeaux TCP. Enfin un agent logiciel associe à tout nouveau flot l'application qui l'a généré au sens de nos critères statistiques.

L'apprentissage des modèles est réalisé hors ligne sur différentes traces. Les résultats d'une campagne de mesure sur une plaque ADSL nous ont été fournis par France Télécom dans le cadre d'un accord de confidentialité, la *groundtruth* ayant été établie par des sondes Otarie effectuant de la DPI. D'autres traces ont été capturées dans notre laboratoire, la groundtruth étant établie grâce à l'outil GT [35]. Les autres traces utilisées sont des traces publiques, par exemple des mesures effectuées sur un réseau du campus de l'université de Brescia [55].

Alors que la validation sur des données GPRS Orange donnait de premiers résultats intéressants une étude plus approfondie des performances a montré que le taux de classification des flots dans le cas multi-classe était plus mauvais que les résultats annoncés par d'autres méthodes de la littérature. Ceci a conduit à s'interroger sur les faiblesses de cette méthode qui est par ailleurs assez bien citée dans la littérature [26].

Une limitation de cette méthode est qu'elle ne peut pas être utilisée pour du trafic encapsulé dans un canal sécurisé, par exemple IPSec, puisque c'est alors tout le paquet IP qui est chiffré ; par contre la critique n'est pas valable dans le cas où IPSec est en mode transport puisqu'alors seule la charge utile du paquet est chiffrée. Une autre limitation est que la méthode se basant uniquement sur les drapeaux TCP, elle est inutile pour le trafic UDP alors que beaucoup d'applications intéressantes utilisent UDP et en particulier RTP/RTCP sur UDP.

Une limitation importante de cette méthode est qu'elle n'utilise que les drapeaux TCP alors qu'on sait

que d'autres descripteurs de niveau paquet, et en particulier les tailles de paquets, sont très informatifs pour la classification. La méthode pourrait être généralisée pour se baser sur des descripteurs mixtes intégrant non seulement certains champs d'entête mais aussi la taille et les espacements entre paquets.

Les premiers échanges de paquets d'une connexion TCP correspondent à la poignée de main initiale SYN, SYN-ACK, ACK. Une majorité des paquets suivants est constituée de simples paquets ACK d'acquittement. L'application ayant généré le trafic n'a pas d'impact sur la poignée de main ou les paquets d'acquittement. Par contre elle en a sur l'utilisation des drapeaux PSH et URG ce qui peut constituer une signature de l'application. En contrepartie une modification du code de l'application pourrait permettre d'échapper au classificateur en positionnant, par exemple, de temps en temps le drapeau PSH à 1.

Des questions se posaient également sur la stabilité de la méthode dans le cas de gigue ou de retransmission. Une possibilité serait de maintenir un état pour chacune des connexions TCP surveillées de façon à gérer correctement le cas des retransmissions ou des paquets délivrés dans le désordre. Mais maintenir un état par connexion est une approche coûteuse en CPU qui doit normalement être réservée aux hôtes terminaux et peut difficilement être mise en place à un niveau d'agrégation de flots élevé dans le réseau à moins de faire un sous-échantillonnage massif au niveau flot. Des expérimentations faites en laboratoire sur notre plateforme de métrologie (Figure 1) avec introduction de diverses perturbations (pertes de paquets, gigue, délais, etc...) sont rassurantes sur la stabilité des performances de la méthode en cas de retransmissions ou de paquets délivrés dans le désordre.

Malgré des premiers résultats prometteurs il s'est avéré que la méthode basée sur les drapeaux TCP présentait des limitations dont nous venons de discuter. Etant donnée la profusion d'articles sur la classification de trafic dans les années qui ont suivi cette publication MineNet 2007 il aurait fallu améliorer la méthode et faire une étude extrêmement fouillée de ses performances dans différents contextes (stabilité spatiale, stabilité temporelle, complexité, consommation, etc...) pour se démarquer par rapport à la communauté. Les moyens humains et le temps disponibles n'ont pas permis d'approfondir cette étude comme on aurait pu le faire.

GROLEAT Tristan, ARZEL Matthieu, VATON Sandrine, Hardware Acceleration of SVM-Based Traffic Classification on FPGA, IWCMC TRAC 2012

Positionnement de l'étude

L'association par des méthodes d'apprentissage automatisé du trafic aux applications l'ayant généré a été largement étudiée depuis les premiers travaux en 2004-2005. De nombreuses méthodes ont été proposées. Ces approches statistiques ou comportementales se basent sur des descripteurs variés au niveau du paquet, au niveau du flot ou des échanges entre un hôte et les autres machines, pairs ou serveurs. Différentes méthodes de classification, le plus souvent supervisées, ont été proposées. Les performances en termes de taux de classification et de matrices de confusion ont été mesurées sur des traces étiquetées obtenues dans différents scénarios, trafic de campus, réseau d'accès d'opérateur, trafic généré en laboratoire.

Alors que les algorithmes eux-mêmes ont été longuement discutés, la question de la conception pratique de ces systèmes a été très peu abordée à l'exception notable de quelques rares contributions récentes [30][34] et de travaux en cours. Tous les opérateurs surveillent en permanence la composition du trafic. Pour cela ils utilisent des sondes qui analysent la charge utile du paquet (DPI). Mais à notre connaissance les méthodes statistiques ou comportementales ne sont pas ou très peu déployées. Pour permettre le transfert technologique il est important de ne pas occulter les aspects liés aux implémentations pratiques de ces méthodes.

Nous travaillons donc à démontrer la faisabilité pratique des algorithmes de classification de trafic par des méthodes statistiques. Nous nous intéressons en particulier à la classification en temps réel du trafic. Nous étudions les performances du système non seulement du point de vue des taux de classification obtenus mais aussi du point de vue des débits traités.

Résumé de l'article IWCMC TRAC 2012

La méthode des machines à vecteur de support (SVM, *Support Vector Machine*) [23] est un algorithme de classification supervisée. SVM est souvent considéré comme l'algorithme le plus performant en

classification de trafic [43][58] et a été adopté par de nombreux auteurs [8][32][30]. SVM transforme un problème de classification non linéaire en un problème linéaire, en utilisant une *méthode de noyau*.

L'objectif est de séparer un ensemble de points dans un espace multi-dimensionnel par des hyperplans pour définir un ensemble de classes. Il est souvent impossible de trouver des hyperplans séparant les points de l'ensemble d'apprentissage. Le principe de la SVM est d'utiliser une fonction noyau pour transformer ces points en des points d'un autre espace dans lequel des hyperplans séparateurs existent. Dans l'espace image la SVM cherche les hyperplans qui séparent les classes et qui maximisent la distance entre les hyperplans séparateurs. Les points qui tombent sur les frontières sont appelés vecteurs de support. Le résultat de la phase d'apprentissage est constitué par la fonction de noyau et les vecteurs de support. La précision dépend du choix de la fonction de noyau, le noyau Gaussien (*Radial Basis Function*, RBF) donnant souvent de bons résultats. Dans la phase de classification les nouveaux points sont classés en fonction du sous-espace auquel ils appartiennent.

Nous considérons comme descripteur de trafic la taille des trois premiers paquets non vides du flot bidirectionnel, en ignorant la poignée de main initiale dans le cas du trafic TCP. La classification des points est obtenue par une méthode SVM. Plusieurs publications font état des bonnes performances de cette méthode [32] [33]. Dans un premier temps nous validons les performances de la méthode sur trois jeux de données correspondant à des scénarios différents : la trace UNIBS-2009 qui contient du trafic de campus de l'université de Brescia [55], une trace d'un laboratoire de recherche Ericsson, et une trace d'une plaque ADSL du réseau Orange France. Les performances obtenues en classification sont satisfaisantes dans les trois cas ; entre 94,5% et 98,5% des flots sont classifiés correctement ce qui confirme les bons résultats annoncés dans la littérature [32] [33].

Puis nous nous intéressons à la classification temps réel des flots au niveau de points de capture dotés d'accélération matérielle. Dans nos études actuelles nous considérons des points de capture à des débits allant de quelques Gb/sec. à quelques dizaines de Gb/sec. Nous avons développé une sonde matérielle réalisant la reconstruction des flots et leur classification par SVM en utilisant des cartes FPGA spécialisées dans le traitement temps réel du trafic. Cette étude a été réalisée dans le cadre de la thèse de Tristan Groléat co-encadrée par Matthieu Arzel (Télécom Bretagne, département Electronique).

Pour nos études sur l'accélération matérielle nous utilisons deux types de cartes : NetFPGA 1G et COMBOv2. La carte NetFPGA 1G dispose de quatre interfaces Ethernet au débit de 1 Gb/sec. La carte COMBOv2 dispose de deux interfaces à 10 Gb/sec. NetFPGA [49] est un projet de plateforme programmable temps réel pour le traitement du trafic lancé par l'université de Stanford et soutenu par différents sponsors dont Xilinx. La carte COMBO et l'environnement de développement associé NetCOPE sont des produits commerciaux de la société INVEA-TECH [39], spin-off de l'université de Brno (République Tchèque) soutenue par Czesnet, l'équivalent tchèque du réseau RENATER d'interconnexion des campus académiques.

Les fonctionnalités de ce point de capture matériel sont de reconstruire les flots à partir des paquets et de classer chaque flot par une méthode SVM se basant sur la taille des trois premiers paquets du flot. Pour la reconstruction des flots nous nous sommes basés sur un projet existant d'implémentation d'une sonde NetFlow sur une carte NetFPGA [59]. Nous avons par ailleurs développé le classificateur SVM en utilisant différentes stratégies : passage en virgule fixe, utilisation de la ROM pour précalculer le résultat de calculs intermédiaires (carrés, exponentiations), parallélisation maximale des traitements concernant les différents vecteurs de support.

Le circuit conçu a été synthéthisé sur un FPGA VIRTEX-5 XC5VTX240. Le résultat de ces expérimentations est que la précision de la classification se dégrade légèrement en raison du passage des calculs en virgule fixe (de 97% des flots classés correctement on passe à 90% des flots). Les délais de rétention des paquets induits par la classification sont faibles : de 57 ns à 254 ns selon les traces considérées. Une accélération considérable du nombre maximum de flots classés par seconde est obtenue grâce au passage au matériel. On passe de 1031 flots/seconde en logiciel (processeur Xeon X5650 à 6 coeurs avec hyperthreading et 12 GO de RAM) à 20 000 flots/seconde en matériel (avec 8 unités de calcul) pour la trace UNIBS-2009, de 4655 flots/seconde (logiciel) à 290 000 flots/seconde (matériel) pour la trace de laboratoire Ericsson, et de 311 flots/seconde à 172 000 flots/seconde pour la trace ADSL Orange France.

PERSPECTIVES

Les perspectives concernant cet axe de recherche s'inscrivent dans la continuité de l'article IWCMC TRAC 2012. Nous souhaitons concevoir un système complet incluant l'apprentissage de modèles SVM fiables pour la classification de trafic et la classification en temps réel et à haut débit des flots de trafic.

Amélioration de la phase d'apprentissage de la classification par SVM

Une faiblesse de la méthode SVM est que, comme pour les autres techniques de classification supervisée, les performances obtenues sont extrêmement sensibles à la façon dont est mené l'apprentissage. On doit donc apporter un soin particulier à l'apprentissage des modèles sur lesquels se base la classification. Par exemple la mise à jour importante d'une application peut changer dans certains cas radicalement le comportement de cette application [57]. Le classificateur ne sera donc plus opérationnel tant qu'une modification du modèle qu'il exploite n'aura pas été réalisée. De plus de nouvelles applications apparaissent régulièrement et la définition même de classes est instable car elle dépend essentiellement de la façon dont les filtres utilisés pour la DPI ont été définis. Le trafic est une cible mouvante qui "mute" au gré des mises à jour de logiciels ou de l'émergence de nouvelles applications et l'apprentissage du modèle SVM doit donc être fait avec le plus grand soin.

Il est nécessaire de concevoir une architecture pour la classification du trafic de façon à autoriser un apprentissage dynamique des modèles des applications. Cette architecture est décomposée en un plan de détection et un plan d'apprentissage. Le plan d'apprentissage doit être conçu de façon à permettre d'adapter dynamiquement les modèles sur lesquels se base la classification réalisée dans le plan de détection. Le principe proposé est d'impliquer un panel d'utilisateurs terminaux. Un point délicat est de motiver cette communauté d'utilisateurs à participer à l'expérimentation. Cela peut s'envisager de plusieurs manières : réduction du prix de l'abonnement Internet, retours réguliers sur des mesures de QoS... Les utilisateurs participant à l'expérimentation installent un agent logiciel sur leurs machines. Cet agent surveille les appels système générés par les applications et les corrèle avec des mesures de trafic au niveau d'une interface réseau, à la manière de l'outil GT de l'université de Brescia . Cette corrélation permet d'étiqueter les flots avec le nom du programme ayant généré le trafic. Les flots étiquetés sont transmis à un collecteur d'apprentissage et stockés dans une base de données. La calibration du modèle de classification SVM est régulièrement réalisée sur ces flots étiquetés.

L'apprentissage des modèles c'est-à-dire la détermination des hyperplans séparateurs doit être effectuée régulièrement sur la base de données des rapports de flots étiquetés. Même si les flots les plus anciens sont effacés régulièrement de la base, les données à traiter sont potentiellement très volumineuses, surtout dans le cas d'un système déployé chez l'opérateur. L'implémentation du mécanisme d'apprentissage doit donc être particulièrement performante. Or la faiblesse principale des méthodes SVM est que la phase d'apprentissage peut être, dans le cas d'instances de grande taille, une tâche particulièrement lourde.

Nous sommes en train d'explorer les possibilités d'accélération de l'apprentissage par parallélisation massive. On peut utiliser au mieux les potentialités des architectures multi-coeur mais cela n'est pas toujours suffisant. Cette étude a fait l'objet d'un stage d'été et d'un projet de dernière année à Télécom Bretagne co-encadrés par Serge Guelton, chercheur associé au département Informatique de Télécom Bretagne. Nous avons choisi une solution basée sur l'utilisation d'une carte graphique. Les cartes graphiques sont des unités de traitement massivement parallèles qui permettent de traiter des problèmes de calcul vectoriel. L'article [16] considère un portage sur GPU de la SVM dans le cas de deux classes. Il existe d'autres travaux récents sur des implémentations CUDA de l'apprentissage SVM comme par exemple [13].

Pour nos tests nous avons fait l'acquisition d'une machine équipée d'un processeur Xeon X5650 à 6 coeurs cadencé à 2.66 GHz avec *hyperthreading*, de 12 Go de DDR3 RAM, d'une carte graphique Nvidia GTX 580 (famille Fermi), exécutant un noyau Linux 2.6.38 et utilisant CUDA 4.0 SDK. Une comparaison de la vitesse d'apprentissage du modèle SVM démontre des coefficients d'accélération intéressants dans certains cas. Nous avons obtenu une accélération par un facteur $\times$ 57 pour l'approche basée GPU par rapport à l'approche basée CPU multi-coeur dans le cas du jeu de données UNIBS-2009 de l'université de Brescia (134000 flots, 8 classes).

Il est maintenant nécessaire de développer l'ensemble des éléments logiciels du plan d'apprentissage, de motiver une communauté d'utilisateurs (étudiants, enseignants-chercheurs, internautes.) à participer

à l'expérimentation, et de tirer les leçons des premières expérimentations sur l'apprentissage de modèles de trafic impliquant un panel d'utilisateurs.

Un autre axe de recherche pourrait consister à ne plus utiliser des approches supervisées comme c'est le cas de la méthode SVM. En effet nous avons vu que les méthodes supervisées peuvent donner de très bons résultats à condition que les modèles soient appris d'une manière extrêmement rigoureuse de façon à s'adapter régulièrement à la réalité du trafic dans le réseau supervisé. Une autre approche serait d'explorer des solutions non supervisées ou partiellement supervisées. La grande majorité des travaux dans le domaine de la surveillance de trafic et en particulier de la classification de trafic utilisent des approches supervisées. Mais certains travaux récents explorent le potentiel des approches non supervisées ou semi-supervisées [14] [15] [6] avec des résultats très prometteurs.

Conception et développement du plan de détection

Il est important de réfléchir à la façon dont notre solution de classification pourrait s'intégrer dans un réseau de production. Ci-dessus nous avons discuté de la conception du plan d'apprentissage. Nous allons maintenant explorer certains aspects liés à la conception du plan de détection.

Deux solutions sont possibles pour le plan de détection : détection centralisée ou décentralisée. Dans l'approche centralisée des points de capture envoient régulièrement à un collecteur centralisé des rapports de flots incluant les informations nécessaires pour la classification (identifiant du flot, taille des trois premiers paquets du flot) ; le collecteur centralisé est un serveur doté d'une puissance de calcul importante qui exécute les algorithmes de classification des flots et stocke les résultats dans une base de données de classification. Dans l'approche décentralisée des sondes dotées de capacités de classification sont positionnées en différents points de capture dans le réseau et remontent les résultats de la classification à un serveur central qui héberge la base de données de classification. Chacune de ces deux approches a des avantages et des inconvénients.

Plusieurs facteurs doivent être pris en compte pour choisir entre l'approche centralisée et l'approche décentralisée. Les opérateurs sont préoccupés par le coût du déploiement de solutions de monitoring dédiées. Les bénéfices de la mesure doivent contre-balancer le coût (CAPEX et OPEX) du déploiement et de la maintenance d'une telle infrastructure. C'est certainement une des raisons pour lesquelles les efforts de la communauté académique n'ont pas abouti à une explosion de solutions implémentées dans les réseaux opérés.

L'implémentation d'un agent de classification au niveau de chaque équipement de réseau nécessiterait le déploiement de sondes dédiées embarquant des fonctionnalités de classification et ce déploiement serait un facteur de coût important pour les opérateurs. Les opérateurs préféreraient utiliser des solutions standardisées comme NetFlow. L'opérateur préférerait collecter des rapports de flot avec des sondes standards et analyser ces rapports au niveau d'un ou plusieurs collecteurs centralisés. Ceci plaide en faveur d'une architecture centralisée pour la classification mais nécessite de doter les collecteurs d'une capacité de calcul très importante.

D'un autre côté centraliser les rapports de flots et utiliser un super-calculateur pour la classification peut consommer une fraction non négligeable de la bande passante si le système est déployé à large échelle. Par contre, dans la solution décentralisée, seuls des statistiques agrégées sur les résultats de la classification effectuée au niveau de chaque point de capture peuvent être collectées ce qui consomme beaucoup moins de bande passante. Un autre facteur en faveur de la solution décentralisée est le délai nécessaire si les rapports doivent être envoyés au collecteur pour classification. Pour certaines applications il peut être nécessaire de classer les flots en temps réel de manière à permettre une réaction appropriées, par exemple l'assignation d'un flot à une classe de service. Dans ce cas là il est nécessaire de réaliser la classification au niveau du point de capture.

La classification doit être faite en temps réel et à des débits élevés ce qui plaide pour des solutions utilisant une accélération matérielle comme celle que nous avons développée [36]. Les axes d'amélioration de notre classificateur matériel consistent à permettre la mise à jour dynamique des modèles en les stockant en RAM plutôt qu'en ROM, et l'utilisation d'un noyau plus adapté aux implémentations matérielles que le noyau RBF (noyau se basant sur des décalages de bits et autres opérations adaptées au matériel) ce qui permettra probablement d'augmenter encore les débits de traitement de manière très significative.

Comparaison avec des approches logicielles optimisées

Le passage au matériel permet une accélération considérable par rapport à une implémentation logicielle utilisant la LibSVM sur une machine puissante (processeur Xeon X5650 à 6 coeurs avec *hyperthreading* et 12 GO de RAM).

Cependant il existe quelques travaux sur des approches logicielles optimisées [30] [34] pour la classification de trafic basée sur une méthode SVM. Il est donc important de se comparer à ces approches logicielles optimisées.

Par ailleurs il existe des implémentations massivement parallèles de la classification par SVM sur carte graphique comme GPUSVM [16] ou CUSVM [13]. En outre des travaux récents comme PacketShader [38] ont démontré la possibilité de paralléliser massivement certains traitements sur les paquets de trafic en exploitant les potentialités de la carte graphique. Dans le cas de PacketShader c'est un routeur logiciel fonctionnant à un débit de 10 Gb/sec. qui a été implémenté.

Il est donc important d'approfondir la comparaison de la classification de trafic basée SVM avec accélération matérielle à des approches basées sur des implémentations logicielles optimisées. Il est important de comparer les performances des approches logicielles et matérielles en termes de débits de traitement. Mais il faut également prendre en compte le facteur consommation électrique dans la comparaison. Les approches logicielles optimisées peuvent offrir des débits intéressants mais le facteur consommation électrique sera probablement nettement en faveur des approches matérielles.

Détection du trafic de streaming

La méthode décrite dans l'article IWCMC TRAC 2012 donne de bons résultats puisque 97% des flots sont classés correctement quand la méthode est implémentée en flottants et que cette précision moyenne est de l'ordre de 90% après passage en virgule fixe. Cependant le trafic de *streaming* semble ne pas être reconnu correctement. La précision est seulement de l'ordre de 60 à 65% des flots pour cette catégorie d'applications. Etant donné l'importance du *streaming* aujourd'hui il est nécessaire d'améliorer la reconnaissance de ce type de trafic. Pour l'instant nous ne comprenons pas les raisons pour lesquelles ce trafic est mal reconnu et nous souhaitons approfondir nos recherches sur ce point.

Génération de trafic à haut débit

Un autre axe est de tester nos sondes matérielles en temps réel sur du trafic produit en laboratoire par un générateur de trafic et sur du trafic de production. Pour obtenir un accès à du trafic de production nous avons positionné une sonde sur un port miroir du routeur du RESEL [4], réseau géré par les élèves et permettant l'accès Internet depuis les résidences universitaires du campus de Télécom Bretagne. Nous avons ainsi accès au trafic généré par 600 étudiants à un débit maximum de 300 Mb/sec. Cette sonde a été mise en place avec l'accord des étudiants, la décision ayant été validée par un vote en Assemblée Générale du RESEL à l'automne 2011. Ceci nous donne accès à un trafic représentatif de résidences universitaires mais à des débits qui restent faibles par rapport à ceux que nous visons. Un projet d'étudiants est en cours avec pour objectif de détecter le trafic P2P sur le RESEL et de mettre en place une politique de blocage adaptée.

Par ailleurs nous souhaitons disposer d'une solution de génération de trafic à haut débit en laboratoire. Les générateurs de trafic comme Spirent AX4000 produisent un trafic réputé être peu réaliste mais nous permettraient de tester nos implémentations sur du trafic haut débit. Pour le moment nous ne disposons pas de générateur commercial de ce type car le coût de ce type de matériel est élevé pour un usage académique. Par contre la société INVEA-TECH, partenaire du projet DEMONS et vendeur de la carte COMBO, nous a donné accès à l'environnement logiciel NetCOPE. NetCOPE s'exécutant sur le processeur embarqué dans le boîtier COMBO permet de générer du trafic à un débit d'environ 5 Gb/sec. La solution implémentée dans NetCOPE se base sur du rejeu de traces sous forme de fichier au format PCAP.

Enfin un projet de dernière année à Télécom Bretagne a été réalisé avec pour objectif de concevoir un générateur de trafic paramétrable sur carte NetFPGA 1G. La solution développée par les étudiants permet de générer du trafic à quelques Gb/sec. à partir d'une spécification statistique des flots dans un

format XML. Ce travail vient d'être finalisé et devrait donner lieu à une publication en workshop.

Références

[1] Internet Measurement Conference 2011. http://conferences.sigcomm.org/imc/2011/.

[2] Passive and Active Measurement Conference. http://pam2012.ftw.at/.

[3] ANR RNRT (projet labellisé fin 2005). OSCAR : Overlay Security, Characterization, Analysis and Recovery.

[4] Association Résel : réseau des élèves de Télécom Bretagne. Bienvenue sur le site du RESEL ! http://resel.fr.

[5] T. Auld, A.W. Moore, and S.F. Gull. Bayesian neural networks for internet traffic classification. *Neural Networks, IEEE Transactions on*, 18(1) :223 –239, 2007.

[6] K. Avrachenkov, P. Gonçalvez, A. Legout, and M. Sokol. Classification of Content and Users in BitTorrent by Semi-supervised Learning Methods. In *Proceedings of the 8th International Wireless Communications and Mobile Computing Conference*, IWCMC TRAC '12, 2012.

[7] F. Baboescu and G. Varghese. Scalable packet classification. *IEEE/ACM Trans. Netw.*, 13 :2–14, 2005.

[8] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti. Abacus : Accurate behavioral classification of P2P traffic. *Elsevier Computer Networks*, 55(6) :1394–1411, 2011.

[9] L. Bernaille, Renata R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36 :23–26, 2006.

[10] CAIDA. CAIDA Data - Overview of Datasets, Monitors, and Reports. http://www.caida.org/data/overview/.

[11] CAIDA. CoralReef Software Suite. http://www.caida.org/tools/measurement/coralreef/.

[12] M. Canini, W. Li, M. Zadnik, and A.W. Moore. Experience with high-speed automated application-identification for network-management. In *Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '09, 2009.

[13] A. Carpenter. Cusvm : A cuda implementation of support vector classification and regression. http://patternsonascreen.net/cuSVM.html.

[14] P. Casas, J. Mazel, and P. Owezarski. Knowledge-Independent Traffic Monitoring : Unsupervised Detection of Network Attacks. *IEEE Network Magazine, Special Issue on "Network Traffic Monitoring and Analysis"*, 26(1), 2012.

[15] P. Casas, J. Mazel, and P. Owezarski. Unsupervised Network Intrusion Detection Systems : Detecting the Unknown without Knowledge. *Computer Communications*, 2012.

[16] Bryan Christopher Catanzaro, Narayanan Sundaram, and Kurt Keutzer. Fast support vector machine training and classification on graphics processors. In *Technical Report No. UCB/EECS-2008-11*. EECS Department, University of California, Berkeley, 2008.

[17] C.Callegari and S.Vaton (Chairs). TRAC'10 : 1st International Workshop on Traffic Analysis and Classification. http://netgroup.iet.unipi.it/trac2010/Home_Page.html.

[18] C.Callegari and S.Vaton (Chairs). TRAC'11 : 2nd International Workshop on Traffic Analysis and Classification. http://netgroup.iet.unipi.it/trac2011/Home_Page.html.

[19] C.Callegari and S.Vaton (Chairs). TRAC'12 : 3rd International Workshop on Traffic Analysis and Classification. http://netgroup.iet.unipi.it/trac2012/Home_Page.html.

[20] CISCO Systems. Cisco Visual Networking Index : Forecast and Methodology, 2010-2015. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[21] Clear Foundation. l7-filter : application layer packet classifier for Linux. http://l7-filter.clearfoundation.com/.

[22] CNNMoney. Apple fact check : 50,000 iPhone apps ? http://tech.fortune.cnn.com/2009/06/10/apple-fact-check-50000-iphone-apps/.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[24] COST Action IC0703. TMA Portal : European Research Portal on Traffic Monitoring and Analysis. http://www.tma-portal.eu/.

[25] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *SIGCOMM Comput. Commun. Rev.*, 37 :5–16, 2007.

[26] H. Dahmouni, S. Vaton, and D. Rossé. A markovian signature-based approach to IP traffic classification. In *MineNet 2007 : ACM Sigmetrics Workshop on Mining Network Data*, pages 29 – 34, 2007.

[27] Dr Peering (Bill Norton). DrPeering.net : The tools and talent needed to make strategic peering decisions. . `http://drpeering.net/`.

[28] Endace. Endace DAG high-speed capture cards. `http://www.endace.com/endace-dag-high-speed-packet-capture-cards.html`.

[29] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, MineNet '06, pages 281–286, New York, NY, USA, 2006. ACM.

[30] A. Este and F. Gringoli. On-line SVM traffic classification. In *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference*, IWCMC TRAC '11, 2011.

[31] A. Este, F. Gringoli, and L. Salgarelli. On the stability of the information carried by traffic flow features at the packet level. *SIGCOMM Comput. Commun. Rev.*, 39, June 2009.

[32] A. Este, F. Gringoli, and L. Salgarelli. Support vector machines for tcp traffic classification. *Computer Networks*, 53(14) :2476 – 2490, 2009.

[33] G.Gomez and P.Belzarena. Early Traffic Classification using Support Vector Machines. In *Fifth International Latin American Networking Conference*, LANC'09, 2009.

[34] F. Gringoli, L. Nava, A. Este, and L. Salgarelli. MTCLASS : enabling statistical traffic classification of multi-gigabit aggregates on inexpensive hardware. In *Proceedings of the 8th International Wireless Communications and Mobile Computing Conference*, IWCMC TRAC '12, 2012.

[35] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and K.C. Claffy. GT : picking up the truth from the ground for Internet traffic. *ACM SIGCOMM Computer Communication Review*, 39(5) :13–18, 2009.

[36] T. Groleat, M. Arzel, and S. Vaton. Hardware acceleration of SVM based traffic classification on FPGA. In *IWCMC TRAC'12 : 3rd International Workshop on Traffic Analysis and Classification*, Limassol, Cyprus, 2012.

[37] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS : Automated construction of application signatures. In *In SIGCOMM 2005 MineNet Workshop*, 2005.

[38] S. Han, K. Jang, K.S. Park, and S. Moon. PacketShader : a GPU accelerated software router. `http://shader.kaist.edu/packetshader/`.

[39] INVEA-TECH. FPGA Solutions Overview. `http://www.invea-tech.com/fpga-solutions/fpga-solutions-overview`.

[40] M. Jaber, R. Cascella, and C. Barakat. Can we trust the inter-packet time for traffic classification ? In *Proceedings of the IEEE International Conference on Communications (ICC)*, ICC '11, 2011.

[41] W. Jiang and M. Gokhale. Real-time classification of multimedia traffic using fpga. In *Proceedings of the 2010 International Conference on Field Programmable Logic and Applications*, FPL '10, 2010.

[42] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc : multilevel traffic classification in the dark. *SIGCOMM Comput. Commun. Rev.*, 35 :229–240, 2005.

[43] Hyunchul Kim, Dhiman Barman, Michalis Faloutsos, Marina Fomenkov, and Kiyoung Lee. Internet traffic classification demystified : The myths, caveats and best practices. In *In Proc. ACM CoNEXT*, 2008.

[44] W. Li, M. Canini, A.W. Moore, and R. Bolla. Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema. *Computer Networks, Special Issue on Traffic classification and its applications to modern networks*, 53 :790–809, 2009.

[45] Y. Luo, K. Xiang, and S. Li. Acceleration of decision tree searching for IP traffic classification. In *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '08, 2008.

[46] Matthew M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-service mapping for qos : a statistical signature-based approach to ip traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 135–148, New York, NY, USA, 2004. ACM.

[47] Martin A. Brown. Traffic Control HOWTO. `http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html`.

[48] A.W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '05, pages 50–60, New York, NY, USA, 2005. ACM.

[49] NetFPGA. NetFPGA : a line-rate, flexible, and open platform for research, and classroom experimentation. `http://netfpga.org/`.

[50] T. Nguyen and G. Armitage. A survey of techniques for Internet traffic classification. *IEEE Communications Surveys and Tutorials*, 10(4) :56–76, 2008.

[51] D. Rossi and S. Valenti. Fine-grained traffic classification with netflow data. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC TRAC'10, pages 479–483, 2010.

[52] C. Rotsos, Jurgen J. Van Gael, A.W. Moore, and Z. Ghahramani. Probabilistic graphical models for semi-supervised traffic classification. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, pages 752–757, New York, NY, USA, 2010. ACM.

[53] SearchTelecom.com. Mobile appliance explosion changes broadband mobile network planning. `http://searchtelecom.techtarget.com/tip/Mobile-appliance-explosion-changes-broadband-mobile-network-planning`.

[54] D.E. Taylor. Survey and taxonomy of packet classification techniques. *ACM Computing Surveys*, 37(3) :238–275, 2005.

[55] University of Brescia. UNIBS : Data Sharing. `http://www.ing.unibs.it/ntw/tools/traces/index.php`.

[56] University of Waikato, WAND Group. WITS : Auckland VI. `http://www.wand.net.nz/wits/auck/6/auckland_vi.php`.

[57] S. Valenti and D. Rossi (directeur). *Dealing with P2P traffics in modern networks : measurement, identification and control*. Thèse de doctorat de TELECOM ParisTech, Paris, 2011.

[58] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 2006.

[59] Zadnik M. and Lhotka L. Hardware accelerated Netflow probe. Tech. Report CZESNET, Prague, 2005.

A Markovian Signature-Based Approach to IP Traffic Classification

Hamza Dahmouni
ENST Bretagne
Department of computer Science
Brest, France
hamza.dahmouni@enst-bretagne.fr

Sandrine Vaton
ENST Bretagne
Department of computer Science
Brest, France
sandrine.vaton@enst-bretagne.fr

David Rossé
France Telecom
Research and Development
Belfort, France
david.rosse@orange-ftgroup.com

ABSTRACT

In this paper we present a real-time automatic process to traffic classification and to detection of abnormal behaviors in IP traffic. The proposed method aims to detect anomalies in the traffic associated to a particular service, or to automatically recognize the service associated to a given sequence of packets at the transport layer. Service classification is becoming a central issue because of the emergence of new services (e.g., P2P, VoIP, Streaming video, etc…) which raises new challenges in resource reservation, pricing, network monitoring… In order to identify a specific signature to an application, we first of all model the sequence of its packets at the transport layer by means of a first order Markov chain. The parameters of that Markov chain are obtained from the analysis of dataset with a large number of connections. Then, we decide which service should be associated to new sequence by means of standard decision techniques (Maximum Likelihood criterion, Neyman-Pearson test). The evaluation of our automatic recognition procedure using live traffic traces demonstrates the feasibility and the excellent performance of this approach.

Categories and Subject Descriptors

C.2.3 [**Computer Communications Networks**]: Network Operations -- Network Management, Network Monitoring

General Terms

Reliability, Security, Management

Keywords

Traffic Classification, Küllback-Leibler Distance, Network Monitoring, Anomaly Detection, Markov Chains.

1. INTRODUCTION

The mobile systems GPRS, their evolution EDGE, and UMTS can act as packet wireless access to existing Internet services. However, Internet services are growing rapidly, especially with the appearance of new wider bandwidth consuming services, such as peer-to-peer (P2P), video, VoIP, as well as an increasing utilization of encrypted services. The evolution of towards a mobile access to these bandwidth consuming services naturally induces the appearance of congestion phenomena in these networks. Network operators are thus constrained to control and to detect the different types of traffic circulating within their networks in order to enforce a policy of differentiated pricing and to establish schemes for differentiated bandwidth sharing.

That is the reason why it is necessary to have some automatic classification methods in order to associate a flow of packets to its class of service (CoS). Indeed, data traffic can be associated to an application or to a category. For example, in QoS management, it is inevitable to classify applications by categories in order to dynamically allocate resources taking into account the applications constraints and requirements.

Another application of service classification is the development of traffic source models. In this case the traffic is classified by applications and then statistical parameters (e.g., the flow size distribution, packet inter-arrival times, etc…) are extracted for each service in order to feed performance models and dimensioning tools.

The application area of traffic identification is large. Some typical examples of applications are presented below:

- association of the CoS to a sequence of IP packets;
- protection of the network against the attacks;
- detection and stopping of any malicious or illicit traffic;
- estimation of the traffic load per service;
- application of an invoicing policy

Service classification has received considerable attention for a few years. Existing methods to traffic classification can be sorted in three categories. The first category is based on the correspondence between the port number and the application type defined by the IANA. The second category is based on the syntactic analysis of the

SIGMETRICS '07 Workshops, June 12, 2007, San Diego, CA.

applicative layers. The last category is using supervised learners such as k-Nearest Neighbors (k-NN), Linear (LDA) or Quadratic (QDA) Discriminant Analysis, Support Vector Machines (SVM)… which inputs parameters are a multivariate statistical description of the traffic (e.g., packet sizes, combined with packet interarrival-times, etc…)

The above mentioned methods have a number of serious limitations as we explain in section 2. The purpose of our study is to produce a new method to service recognition on multi-services networks that bypasses these limitations. In fact, contrary to existing methods which are based on syntactic or statistical aspects of the applications, our method takes into consideration the functional aspects of the applications. Indeed each application differs by the communication procedures and by the protocols stack employed. Basing our study on the functional aspects of the applications we are able to define a specific signature for each service.

More precisely we consider in this paper TCP traffic only since it is the dominant traffic today. The observations consist in a sequence of the first control packets - as combination of flags (SYN, ACK, PSH, RST, FIN) - for each TCP connection. This sequence is statistically significantly different from one application to another. This is captured through the definition of a first order Markov chain model; the parameters of the model (state transition probabilities) are different for each application.

The state transition diagram is obtained for many applications (HTTP, HTTPS, Telnet, Mail, etc...) from the analysis of a large traffic dataset collected on the Orange France network. This defines a specific signature for each application. Then, new TCP connections are associated to a given application by a decision rule which is based on the likelihood criterion and on a Neyman-Pearson test. The potential of this new method to service classification is excellent; this is in particular demonstrated by Receiver Operating Characteristics (ROC) curves.

The rest of the paper is organized as follows. Section 2 proposes a taxonomy of the mostly-known service classification methods and discusses their limitations; we also introduce the main steps of our approach. Section 3 presents a detailed description of our traffic classification method and of its main steps (traffic collection, Markovian signature definition, decision rules to classify a new traffic). The performance of the proposed method is evaluated in section 4. The last section provides a conclusion.

2. A TAXONOMY OF SERVICE CLASSIFICATION METHODS

Traffic classification and anomaly detection in IP traffic have gained considerable interest in the past years because of the emergence of new applications (P2P, VoIP, etc…) with specific constraints and the possibility to access these services from a mobile terminal with the HSDPA technology for example. This raises new challenges for resource sharing and differentiated pricing since these new services are bandwidth consuming. In this section we propose a taxonomy of the existing methods that have been published in the literature and we discuss the limitations of each category of methods. We also introduce a new method which makes use the functional aspects of the applications and bypasses the limitations of the existing methods.

2.1 Port-based classification

Traffic classification by port number is the simplest and most traditional method. It consists in identifying the application type from the port number in the transport layer (TCP or UDP). The correspondence between the port number and the required application is defined by the IANA (Internet Assigned Numbers Authority) [6].

However, this method has several limitations. Firstly, the association of the port number to an application is not always possible for different reasons: the port numbers are not defined by the IANA for all the applications, especially for the new ones, for instance P2P applications. Moreover, several TCP implementations employ client ports in the registered ports range; this could erroneously classify the connection as generated by the application associated with this port. The server ports can also be dynamically assigned; for example, the FTP passive mode allows the dynamic negotiation of the port number used for the data transfer. This number is given during the initial control connection (FTP control using server port 20). Another limitation is that different services can be encapsulated in well-known applications, for example streaming or chat on HTTP, etc. Finally, various services (real time, interactive, elastic) with different levels of QoS can use the same port number. For all these reasons, there are serious limitations to this approach to service classification.

2.2 Payload-based classification

An alternative to port number-based classification is to employ a syntactic analysis of the IP datagram payload. This method consists, more precisely, in seeking deterministic character strings in the applicative layer of the packets. Indeed, this approach requires a precise knowledge of the application-layer data as well as the format of its packets. In practice, it is necessary to seek character strings which identify an application. For example "http/1." corresponds to application HTTP, 0xe319010000" corresponds to eDonkey services [8][14].

This technique is an online and quasi-deterministic classification method. But it has also many limitations. Firstly, a description of a characters string that would identify an application or a given version of this application is not always available. In the case of secured applications the applicative payload is encrypted and this method becomes inapplicable. The packet format and type are different from an application to another. Therefore, this information is not easy to extract, since it depends on the adjacent protocols employed. Moreover, data services with different constraints and requirements can be encapsulated

in traditional applications (e.g., the streaming on HTTP, video on demand, etc…).

2.3 Statistical-based classification

Various types of statistical classifiers were proposed in order to classify the data traffic in terms of applications. We can mention, for instance, LDA, k-NN and SVM methods. Traditionally, the researches were oriented towards a statistical characterization of the different applications (e.g., [1], [3], [7]) in order to define input parameters to, for example, dimensioning models or tools. These studies supposed that the different applications were identified unambiguously on the considered multi-services network. The goal was to obtain statistical characteristics by applications. The converse problem is now gaining importance because of the current increase in the number and variety of Internet applications. The problem is now to associate a given flow, characterized by some statistical parameters, to the corresponding service. This classification can be performed by statistical learning methods.

Initially, C. Dewes et *al*. [4] examined the statistical properties of "chat" traffic and then devised a heuristic to isolate "chat" traffic from a large amount of other traffic. M. Roughan et *al.* [13] extended [4] by proposing a more rigorous classification approach: they represented each application by a statistical signature formed by different traffic descriptors and then used two supervised learning techniques (k-NN and LDA) in order to find a correspondence between different CoS (interactive, streaming, etc...) and statistical traffic descriptors.

This approach consists in collecting statistical characteristics from certain traffic aggregates defined by the triplet: {IP address, port number, TCP connection}. The authors associate statistical descriptors (e.g. average packet size, connection duration, etc...) to each TCP connection. Then LDA and k-NN methods are used to define correspondence rules between the set of statistical descriptors that characterize each aggregate and the class of service. However, the results show high error rates of these methods (57% for LDA and 14% for 3-NN and 5-NN).

R. Kwitt et *al*. [9] propose a similar study to [13] using different learning methods, in particular Linear (LDA), Quadratic (QDA), Flexible (FDA) and Mixed (MDA) Discriminant Analysis methods. A. Moore et *al*. [11] used a naïve Bayesian classifier in order to classify Internet traffic into categories. In that case, the characterization was done at the flow level and as much as 248 traffic descriptors were used in order to describe each. The obtained classification rate was 65% if a naïve Bayesian method was used, but it reached 95% if two additional improvements were introduced, the Kernel Density Estimation (KDE) and the Fast Correlation-Based Filter (FCBF). The kernel density estimator was used to estimate the probability density of each descriptor whereas the fast correlation-based filter was used to reduce the dimension of the descriptors' space and to choose the relevant descriptors.

Statistical-based classification suffers from a number of limitations. The first limitation is the performance of the method (classification rate) is generally lower than what can be obtained with other methods. The performance is not stable since it depends mainly on the statistical descriptors chosen (more than on the chosen classifiers) contrary to deterministic approaches such as payload-based methods. The selection of relevant statistical descriptors is consequently very important. The last limitation is that it would be very easy for a malicious user to bypass some filters based on this approach and gain access to some applications that are for example prohibited on a given network. Indeed traffic descriptors such as packet length for example can be maliciously modified (by padding) in order to mislead the classifier.

2.4 A Markovian signature-based approach

In spite of the diversity of work completed on this subject, research in this field continues to be relevant, as the results already obtained are still subject to improvement. In this context, we propose a new original method for anomalous behavior detection and services classification in fixed and mobile IP networks. The goal of this proposal is twofold. The first goal is to detect anomalies in the traffic corresponding to a given application. The second goal is to mitigate certain limitations of the existing methods to service classification. The proposed approach aims to fill the needs of a network operator; for example compared to the other methods, it will allow network operators to:

- detect an abnormal behavior of an application if it is encapsulated in other well-known applications (for example, streaming or "chat" over HTTP);
- detect an illicit or malicious traffic using protocols and ports classically used by well-known applications;
- identify encrypted services (e.g. encrypted P2P, etc.);

Our method is making use of the protocol and functional aspects of the applications, contrary to existing methods which are based either on syntactic aspects or on statistical aspects or port numbers. Indeed, during a connection, an application exchanges a typical sequence of control packets with a remote host (client or server). Our method can be decomposed into two parts:

- Firstly, we model this sequence of symbols by means of a first order Markov chain. The various types of control packets exchanges (e.g. SYN, ACK, PSH-ACK, SYN-ACK, etc...) make up the states space of this Markov chain. The transition matrix of the Markov model, made up of the transition probabilities between states, identifies a "signature" for this application.

  Traffic events modeling by Markov chains has been employed by N. Ye et *al.* [14]. They modeled the Unix events sequence in order to define a signature by machine, used for Cyber-attacks detection in the traffic. In related work, J.M Estevez-Topiador et *al.* [5] proposed a Markovian modeling of TCP segments for

detecting anomalies in the usage of protocols in computer networks.

- Secondly, we use decision theory in order to detect by which application the observed sequence of packets was produced. Our goal is to use decision theory (Maximum Likelihood decision, Neyman-Pearson tests, etc...) to automate services recognition and traffic classification. In the next section, we present a detailed mathematical description of our method.

3. METHOD DESCRIPTION

In order to reach our objective we have investigated GPRS data traces collected on the live Orange France network. The measurements were collected on two different G_b interfaces (interface between SGSN and PCU entities) during a ten day period (about 5 hours per day, between 11 a.m and 4 p.m). The frames captured at the G_b interface contain the TCP and UDP/IP header information (packet size, port number, transfer direction TCP flags, etc…) and a timestamps which indicates the arrival time of packets with an accuracy of millisecond. This information allows the parameterization of applications running within the network.

3.1 Markovian signature definition

The identification of the Markov model associated to an application can be decomposed into three steps. The first step (*i*) consists in identifying the states space, this amounts to determining the various types of control packets used by that application. The second step (*ii*) consists in reconstructing the original order of the packets in a flow; this amounts to reorganizing the flows according to their activation order in the session and the packets according to their emission order ~~of~~ in the flow. The last step (*iii*) consists in estimating the Markov Chain parameters (state transition probabilities).

***i*)** In this study we take only into account the TCP traffic. The control packet type is then given by the TCP "flag" field. This flag contains 6 bits {SYN, ACK, PSH, RST, URG, FIN}, each bit is positioned to 1 or 0. The packet type is a binary combination of the values of these bits. For example, a packet is "SYN" if SYN field is positioned to 1 and all the others to 0, it is "SYN-ACK" if SYN and ACK fields are positioned to 1 and all others to 0. We consider that each packet type represents a state of the Markov chain process. Then, we define the states space from equation (1) in order to have numerical values for the states.

$$E = SYN + 2.ACK + 4.PSH + 8.RST + 16.URG + 32.FIN \quad (1)$$

ii) During a session, the application activates one or more flows in parallel or in series to complete its transfer. The number of flows depends in particular on the application nature, and more precisely it depends on the version of the application.

As the IP data traffic is bursty, it is necessary to reorder the flows and the packets inside a session. This regrouping of packets is a key stage of the treatment, since it makes possible the reconstitution of various TCP serial connections. We assume that the order of the packets composing the same flow is not changed between their emission and their reception. In fact, as the control packets have small sizes, we assume that they generally arrive in the order of their number of sequence. So we reconstitute the order of the flows, and we affect the states corresponding to each packet according to the equation (1).

***iii*)** In the last stage, we model this sequence of packets by a Markov chain $(X_n)_n$ with states space $S:=\{E_1...,E_s\}$. We suppose that connections in the same session are independent, i.e. connections are supposed to be infinite. Meanwhile the end of a flow and the beginning of the following flow are considered to belong to the same connection. This assumption is taken into account in order to lengthen the size of the observed sequence and to increase the algorithm precision.

The transition probability matrix of this chain is noted:

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1s} \\ P_{21} & P_{22} & \dots & P_{2s} \\ \dots & \dots & \dots & \dots \\ P_{s1} & P_{s2} & \dots & P_{ss} \end{bmatrix}, \; P_{ij} = \Pr\{X_{n+1} = j / X_n = i\} \; \forall i, j \in E$$

The P_{ij} element represents the probability that a control packet of type *j* succeeds to a control packet of type *i*. The transition probabilities and the steady-state probabilities for each application are estimated on the collected data traces.

3.2 Decision rules

During its connection, one application exchanges several control packets with the other final entity. Consequently, the client (or server) can be regarded as an information source, and can be modeled by a stochastic process X with discrete states space $S=\{E_1,E_2,...,E_s\}$. Furthermore, we assume that X can be modeled by a homogeneous first order Markov chain.

The entropy of this source is defined, by C. Shannon, as the quantity of average information associated with the realization of each symbol. The expression of the entropy of the source in the case of independent symbols is well known. The notion of entropy can be generalized to the case of dependent symbols. In particular if the source is modeled by a Markov chain with transition matrix $P=(P_{ij})_{i,j}$ and stationary distribution $\pi=(\pi_i)_i$, the entropy per symbol of this source is [2]:

$$H(MC) = -\sum_{i,j} \pi_i \, P_{ij} \, \log\left(P_{ij}\right) \quad (2)$$

Moreover, the Küllback-Leibler distance between two Markov chains models, MC_0 Model with parameter $\{p^0, P^0\}$ and stationary distribution π^0, and MC_1 Model with parameter $\{p^1, P^1\}$ and stationary distribution π^1 is given by [2]:

$$D(MC_1 // MC_0) = \sum_{i,j} \pi^1(i) P^1(i,j) \log\left(\frac{P^1(i,j)}{P^0(i,j)}\right) \quad (3)$$

The KL distance between two Markov chains models representing two applications is close to 0 if the two applications have similar functioning (e.g., for HTTP versus HTTPS applications case). Otherwise, this distance is obviously large (e.g., HTTP versus Telnet or FTP applications case).

We assume that the traffic mix is composed of M applications. Each application k is represented by a Markovian model MC_k. Let $H(MC_k)$ be the entropy of this model. We note $D(MC_k//MC_j)$ KL-distance between MC_k and MC_j models. And we note θ the parameters of the Markovian model MC. Let $(y_1,\ldots,y_T)$ denote the observed samples. Its log-likelihood under parameter θ:=$\{p,P\}$ is:

$$\begin{aligned}\frac{1}{T}\log p(y_1,\ldots,y_T;\theta) &= \frac{1}{T}\log p(y_1) + \frac{1}{T}\sum_{t=2}^{T}\log P(y_{t-1},y_t) \\ &= \frac{1}{T}\sum_{i=1}^{s}\log p_i \, 1_{\{y_1=i\}} + \frac{1}{T}\sum_{i,j}\log P_{ij}\, n_{ij}\end{aligned} \quad (4)$$

where 1_A is the indicator function of event A that takes the value 1 if A is true and 0 otherwise. n_{ij} represents the number of transition observed from state i to the state j ($n_{ij} = \sum_{t=2}^{T} 1_{\{y_{t-1}=i, y_t=j\}}$). The empirical transition probability from state i to the state j is given by:

$$\hat{P}_{ij} = \frac{n_{ij}}{\sum_j n_{ij}} = \frac{n_{ij}}{\sum_{t=2}^{T} 1_{\{y_{t-1}=i\}}} = \frac{n_{ij}}{\hat{\pi}_i (T-1)} \quad (5)$$

where $\hat{\pi}_i$ is the empirical probability distribution of state i. Thus, the equation (4) can be expressed:

$$\begin{aligned}\frac{1}{T}\log p(y_1,\ldots,y_T;\theta) &\simeq \sum_{i,j}\pi_i P_{ij}\log\frac{P_{ij}}{\hat{P}_{ij}} + \sum_{i,j}\hat{\pi}_i\hat{P}_{ij}\log\hat{P}_{ij} \\ &\simeq -D\left(MC(\hat{\theta})//MC(\theta)\right) - H\left(MC(\hat{\theta})\right)\end{aligned} \quad (6)$$

The quantity $H(MC(\hat{\theta}))$ does not depend on θ. Thus maximizing the log-likelihood (5) amount to minimizing the KL distance between the parametric model with parameter θ and the empirical model with parameter $\hat{\theta}$:

$$\underset{\theta}{\text{ArgMax}}\left(\frac{1}{T}\log p(y_1,\ldots,y_T;\theta)\right) = \underset{\theta}{\text{ArgMin}}\left(D\left(MC(\hat{\theta})//MC(\theta)\right)\right) \quad (7)$$

Our decision rule uses Maximum Likelihood (ML) decision. Indeed ML estimation is asymptotically efficient: when the length of the observed sequence is very large the ML estimate reaches the Cramer-Rao Lower Bound which is a lower bound on the variance of any unbiased statistical estimator, based on Fisher information.

But this optimality is asymptotical that is to say that the length of the observed sequence (number of packets) should be very large for the ML decision to be optimal. In practice we want to take a decision after having observed a few packets only (the order of 10 packets).

The Neyman-Pearson test is optimal in the sense that it is the most powerful test in the case of two (simple) hypotheses. This means that for a given level of false alarm the Neyman-Pearson test is the test with maximum power. This optimality result holds whatever the length of the observed sequence but the Neyman-Pearson test is applicable to the case of a decision between two hypotheses only (in our case, two Markov chain models).

Our decision rule combines Maximum Likelihood estimation and the Neyman-Pearson test. It can be decomposed into two steps:

(i) The first step consists in selecting the two applications which are the most likely, that is to say the two Markov Chain models for which the evaluation of expression (6) for the observed sequence is maximal.

(ii) The problem is then to make a decision between these two models. In a second step a decision is made between these two models using a Neyman-Pearson decision criterion.

4. PERFORMANCE EVALUATION

The performance of a decision test (case of two hypotheses H_0 and H_1) is estimated by the pair (α,β) (false alarm probability, detection probability). In our case each hypothesis (H_0 or H_1) corresponds to a specific application. α (false alarm) is the probability of deciding H_1 when H_0 is true; and β (detection probability or "power" of the test) is the probability of deciding H_1 when H_1 is true.

The ROC-curves plots β as a function of α (different decision thresholds ε correspond to different values of the pair (α, β)). Our goal is to decide how many packets T one should observe in order to be able to take a reliable decision (small false alarm α and high power β).

That is why we plot these ROC curves for various values of T, and for various pairs of applications Figure 1 presents ROC-curves for application HTTP (H_0 hypothesis) versus Telnet (H_1 hypothesis). Figure 2 illustrates ROC-curves for application HTTP (H_0 hypothesis) versus HTTPS (H_1 hypothesis). The decision is good when the power is close to 1 for a small false alarm.

From these various curves, we observe that for applications HTTP and Telnet the decision can be made with a strong power and a weak false alarm starting from only a few number of packets (6 or 10 packets). We point out here that the KL distance between HTTP and Telnet is 0.32 which means that these two applications are

statistically significantly different and that it is very easy to discriminate between these two applications. The power of the test indeed depends on the value of the KL distance between the two models [12].

On the other hand, for HTTP and HTTPS applications, a good decision with a weak false alarm is not possible if the number of observed packets is small (of the order of 10 for example) but it becomes possible to take a reliable decision if the number of observed packets is larger (say, larger than 50). We also point out here that the KL distance between HTTP and HTTPS is 0.05 (much smaller than the KL distance between HTTP and Telnet). This obviously stems from the fact that HTTP and HTTPS have very similar protocol and functional aspects.

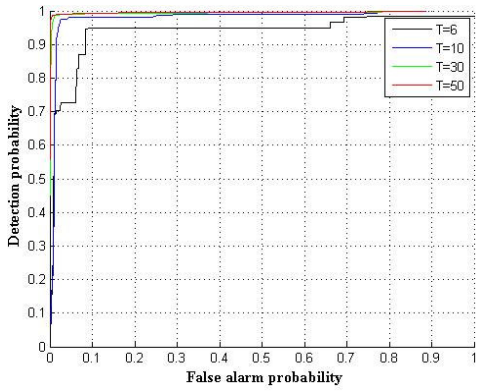


Figure 1. ROC-curves rates for different sequence sizes (T)
H_0:=HTTP and H_1:= Telnet

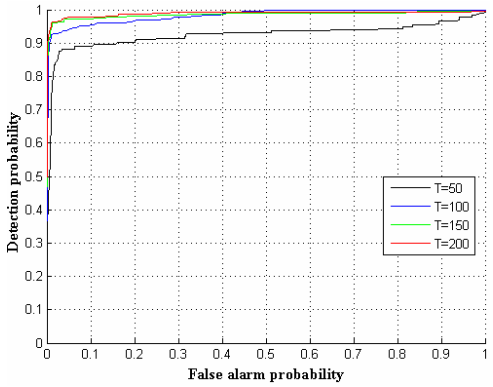


Figure 2. ROC-curves rates for different sequence sizes (T)
H_0:=HTTP and H_1:= HTTPS

5. CONCLUSION

We proposed in this paper a new method of automatic services classification based on a Markov signature of the functional aspects of the applications. We used the Neyman-Pearson test and the likelihood criterion for defining a decision rules. Large traffic datasets from the Orange France operational network were used in order to learn the Markov chain signature of each application and to evaluate the performance of the method. The objective of this method is to automate service recognition as well as supervision of a particular service from the online processing of traffic data. This new process will make it possible to mitigate certain limitations of the existing methods such as port-based, statistical-based or syntax-based service recognition. The evaluation of this method on live Orange France traffic traces demonstrates the very good performance of the method.

6. ACKNOWLEDGEMENT

The authors would like to thank B. Morin (France Telecom Research and Development) as well as the persons in Orange France who provided the necessary measurements.

7. ADDITIONAL AUTHORS

Lionel Fillatre, ENST Bretagne, Signals & communication Department, Brest, France (lionel.fillatre@enst-bretagne.fr)

8. REFERENCE

[1] J. Charzinski. HTTP/TCP connection and flow characteristics", *Performance Evaluation*, vol.42, pages 149-162, 2000.

[2] T. M. Cover, Joy A. Thomas. Elements of Information Theory. Wiley & Sons, 2006.

[3] H. Dahmouni, D. Rossé, B. Morin, S. Vaton. Impact of Data Traffic Composition on GPRS Performance. *ITC'19*, 2005.

[4] D. Dewes, A. Wichmann, A. Feldmann. An analysis of Internet chat systems. *ACM SIGCOMM IMC*, 2003.

[5] J.M. Estevez-Tapiador, P. Garcia-Teodoro, J. E. Diaz-Verdejo. Stochastic Protocol Modeling for Anomaly Based Network Intrusion Detection. *IWIA'03*, 2003.

[6] IANA: http://www.iana.org/assignments/port-numbers

[7] R. Kalden, T. Varga, B Wouters, B. Sanders. Wireless Service Usage and Traffic Characteristics in GPRS networks. *ITC' 18*, 2003.

[8] T. Karagiannis, A. Broido, N.Brownlee, K. Cluffy, Faloutsos M., "Is P2P dying or just hiding. *IEEE Globecom* 2004.

[9] R. Kwitt, T. Strohmeier, U. Hofmann. Machine Learning for Perceptual QoS. *CISTA*, 2005.

[10] A.W Moore, K. Papagiannaki. Toward the accurate Identification of Network Applications. *PAM'05* , 2005.

[11] A.W. Moore, D. Zuev, "Internet Traffic Classification Using Bayesian Analysis techniques", *ACM SIGMETRICS*, 2005.

[12] K. Nakagawa, F. Kanaya. On the Converse Theorem in Statistical Hypothesis Testing for Markov Chains. *IEEE Transactions on Information Theory*, 1993.

[13] M. Roughan, S, Sen. O. Spatscheck, N. Duffield, Class of Service Mapping for QoS: A Statistical Signature Based Approach To IP Traffic Classification. *ACM Sigcomm IMC*, 2004.

[14] S. Sen, O. sparscheck, D. Wang. Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. *13th international conference on WWW*, 2004.

[15] N. Ye, Y. Zhang, C.M. Borror. Robustness of the Markov-Chain Model for Cyber-Attack Detection. *IEEE Transactions on Reliability,* Vol. 53, N°1, 2004.

Hardware Acceleration of SVM-Based Traffic Classification on FPGA

Tristan Groléat
Télécom Bretagne
Brest, France

Matthieu Arzel
Télécom Bretagne
Brest, France

Sandrine Vaton
Télécom Bretagne
Brest, France

***Abstract*—Understanding the composition of the Internet traffic has many applications nowadays, mainly tracking bandwidth consuming applications, QoS-based traffic engineering and lawful interception of illegal traffic. Although many classification methods such as Support Vector Machines (SVM) have demonstrated their accuracy, not enough attention has been paid to the practical implementation of lightweight classifiers. In this paper, we consider the design of a real-time SVM classifier at many Gbps to allow online detection of categories of applications. Our solution is based on the design of a hardware accelerated SVM classifier on a FPGA board.**

I. Introduction

Traffic classification is the task of associating network traffic with the generating application or category of application. In some cases operators would appreciate knowing which application packets belong to in order to better engineer the traffic, charge their customers, etc. All operators are continuously tracking the composition of traffic per category of applications. They are analyzing trends and tracking the emergence of new bandwith consuming applications. Quality of Service (QoS) solutions which segregate traffic into classes and give them different priorities also require the ability to associate traffic to applications since the applications do not tag their traffic by themselves. Traffic classification is also useful for differentiated charging and for Service Level Agreements (SLA) verification. Lawful Interception of illegal traffic makes also mandatory for Internet Service Providers (ISP) to analyze their customers' traffic and recognize some illegal or critical traffic.

Traffic classification is a challenging task for several reasons. First, operators have to deal with a huge amount of traffic to analyze as reported by the annual CISCO report [1]. Lightweight traffic classification techniques are made necessary by the volume and rate of traffic that must be analyzed. Second, traditional techniques for traffic classification have some limitations. Traditionally traffic could be classified either by the analysis of port numbers or by Deep Packet Inspection (DPI). Port-based classification is not always reliable since many applications change dynamically their port number or hide themselves behind well-known ports belonging to other applications (e.g. port 80 for HTTP). DPI techniques recognize specific character strings in the packet payload. These techniques can be evaded by applications which cipher their traffic. They are moreover under distress when a large number of signatures have to be recognized in a high bit rate traffic.

This claims for the design of lightweight techniques for traffic classification that do not rely on port numbers or DPI. There has been a large body of litterature on traffic classification [2]–[15] which is an evidence of the interest of the academics towards this topic. A few surveys are available, for example [16]–[18]. Our contribution to this already deeply investigated subject focuses on the architectural design of some ligthweight traffic classification techniques. The performance of different classification techniques has been deeply investigated in terms of the obtained classification rates (i.e. % of flows which are correctly associated to the generating application). But in spite of the plethora of litterature about traffic classification the question of how these methods must be implemented *in practice* to enable online traffic analysis has not received enough attention. There are a few studies that investigate the impact, for example, of packet or flow level subsampling [19] or feature selection [20] on classification accuracy. In [21], authors study the performance of a software version of the SVM algorithm and optimize the processing time in order to deal with high-speed links. But in general there is a lack of litterature about boosting lightweight traffic classification algorithms with hardware and/or software acceleration techniques.

We study the real implementation of a lightweight classification algorithm with a high-performance hardware accelerated solution. We take as a baseline the well-known Support Vector Machine (SVM) [22] which separates flows in a virtual space by hyperlanes. Flows are described by simple packet level features, in that case the size of the first three data packets in the flow [23] [24]. As any supervised classification method, the SVM algorithm consists of two main phases: *a training phase* and *a detection phase*. During the training phase, the algorithm starts from a learning trace labelled with categories of applications and computes the classification model namely the separating hyperplanes. Using this model, the detection phase decides of the category of application of new flows.

We consider a hardware accelerated implementation of the detection phase of the SVM algorithm. As the rate of traffic to be classified can be larger than tens of Gb/sec. in access and core networks, one needs to accelerate the detection algorithm to perform real time classification. We implement a SVM classifier on FPGA to boost the performance and adapt it to high rate capture points. To validate our contributions,

we consider datasets generated in real networks and conduct extensive experiments.

II. Validation of SVM based traffic classification

A. Background on Support Vector Machine (SVM)

SVM [22] is a supervised classification algorithm. It transforms a non linear classification problem into a linear one, using a so called "kernel trick". Given a set of sample points in a multi-dimensional space one would like to separate them by hyperplanes, thus defining different classes. It is often impossible to separate sample points of different classes by hyperplanes and the separating surface is extremely difficult to compute. The idea of SVM is to map, by means of the kernel function, training points to a transformed space where it is possible to find separating hyperplanes. In the target space SVM must find the hyperplanes which separate points belonging to different classes and have a maximum distance between points of both classes and the separating hyperplane. The output of the training phase is made up of the parameters of the kernel and a set of support vectors x_i that define the separating hyperplane. During the detection phase SVM simply classifies new points according to the subspace they belong to.

SVM is often regarded as the best performing algorithm for traffic classification [16] [20] and has been adopted by several authors [15] [24] [21]. The accuracy depends on the selection of the kernel functions where Radial Basis Function (RBF) kernels usually give good results. We use in our implementations the LibSVM [25] library, which is an integrated software for support vector classification allowing multiclass classification, learning, cross-validation and different kernel functions. LibSVM implements different algorithms for applications of SVM to classification, to distribution estimation and to regression problems. Several algorithms exist for SVM-based classification. We have used the C-Support Vector Classification (C-SVC) algorithm [26].

Let us assume that we have a set of training points $x_i \in \mathbb{R}^n, i = 1, \ldots, l$ in two classes and a set of indicator values $y_i \in \{-1, +1\}$ such that $y_i = +1$ if x_i belongs to class 1 and $y_i = -1$ if x_i belongs to class 2. Let us also assume that we have selected a function ϕ such that $\phi(x_i)$ maps training point x_i into a higher dimensional space.

The training phase searches for an hyperplane that separates points $\phi(x_i)$ belonging to classes 1 and 2. The criterion is to maximize the distance of misclassified points to the separating hyperplane. The direction of the separating hyperplane is defined by a vector $w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i)$ where only a few of coefficients α_i are non null. Non null coefficients define the so-called support vectors which characterize the separating hyperplane. The equation of the separating hyperplane is given by $w^T \phi(x) + b = 0$ that is $\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b = 0$ where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the so called "kernel" function.

In the detection phase, any new point x is classified according to the following decision function:

$$\text{sign}(w^T \phi(x) + b) = \text{sign}(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b) \quad (1)$$

x is classified into class 1 if $w^T \phi(x) + b$ is positive and into class 2 if $w^T \phi(x) + b$ is negative.

In this article we use the Radial Basis Function kernel as good results are often obtained with this kernel:

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) \quad (2)$$

From this simple two-class SVM problem, one can easily deal with multi-class SVM classification problems. A usual approach is the so called "one versus one" (1 vs 1) approach. In this approach $\frac{n(n-1)}{2}$ two-class SVM problem are considered, one for each pair of classes. A training phase is performed for each two-class problem thus producing $\frac{n(n-1)}{2}$ separating hyperplanes. Each new point is then classified according to each of those two-class classification problems. The final decision is taken on the basis of a majority vote, that is to say that the new point is allocated to the class which has obtained the highest number of votes.

B. Accuracy of the SVM algorithm

In order to assess the accuracy of the SVM-based classifier we have performed validation over three different datasets. The learning and detection phases have been performed using the libSVM library [25]. The traffic descriptor that is used as input to the SVM classifier is made up of the size of the first three non empty packets of each flow, where a flow is defined as a set of packets with identical 5-tuples (IP Src adress, IP Dest adress, Src port, Dest port, protocol) [23] [24].

We have used for validation three datasets with groundtruth. The groundtruth identifies the application that has generated the traffic flow. It has been obtained either by Deep Packet Inspection (DPI) with for example Linux L7-filter [27] or by using a tool such as GT [11].

The characteristics of the three traffic traces used as benchmarks are listed in Table I. Those three traces correspond to three very different scenarios: campus network, laboratory environment and residential access network. As a consequence the composition of traffic is significantly different from one trace to the other.

1) The FT (France Telecom) dataset has been provided by France Telecom under the terms of a Non Disclosure Agreement. Traffic has been dumped on one geographical zone of an ADSL France Telecom access network and groundtruth has been established by DPI.
2) The Ericsson dataset corresponds to some trafic that has been generated in a laboratory environment of Ericsson research.
3) The Brescia dataset is a public dataset [11]. It corresponds to some traffic captured on a campus network. The groundtruth has been obtained with the GT tool.

The definition of classes is not universal. It mainly depends on the filters that have been defined for packet payload

Trace label	Network of capture	Bytes	Flows	Flows with a known classification	Capture mean rate (kb/s)
Ericsson	Local Area Network at an Ericsson Laboratory	1 755 816 952	39 056	12 858	315.18
Brescia	Campus trace generated at University of Brescia, Italy	746 850 665	153 237	76 182	1 042.9
FT	DSL Link of France Telecom	1 041 481 214	1 065 836	428 794	3 383.1

TABLE I
TRAFFIC TRACES AND THEIR PROPERTIES

inspection (DPI). In order to enable a comparison between traces we have merged applications into different categories that are listed in Table II.

Class label	Class name
1	Web
2	P2P download
3	Direct download
4	Streaming
5	Game
6	Mail
7	Instant messaging
8	Distant control

TABLE II
TRAFFIC CLASSES

Table III provides figures of traffic classification accuracy for each of the three traces that is to say the overall percentage of flows which are correctly classified.

Trace	FT	Ericsson	Brescia
Accuracy (in %)	94.43	98.53	97.41

TABLE III
ACCURACY OF THE SVM ALGORITHM

A global accuracy figure is usually not considered as sufficient to demonstrate the performance of a classifier. Some classes could be frequently misclassified with not much impact on the global figure if only few flows correspond to those classes. A usual representation of results is given by the confusion matrix. Here we provide in Figure 1 the accuracy per category of applications, that is to say the percentage of flows of each category of applications that has been accurately classified.

As one can see from this figure, the accuracy of the SVM algorithm differs from one category of applications to another and from one trace to another. The proportion of a category of applications in a trace impacts the ability of the SVM algorithm to detect it. For example, as class 1 (Web) is present with a good proportion in all three traces, the accuracy of the detection is high. However, as class 4 (Streaming), is almost absent in the three traces it has the worst classification accuracy. Another reason for the low classification rate of Streaming traffic might be that the size of the first packets is not an accurate traffic descriptor for this category of application.

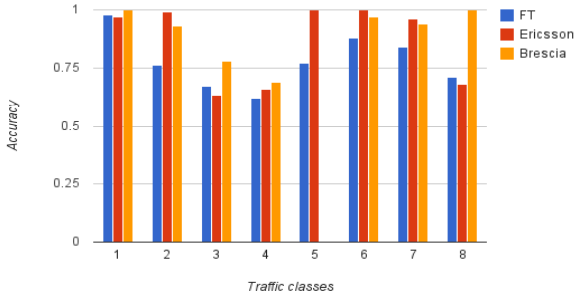


Fig. 1. Accuracy per traffic class

III. TOWARDS ON-LINE TRAFFIC CLASSIFICATION

A. Requirements

In what follows the implementation of on-line SVM traffic classification is studied. In our scenario, probes are located on an operator access or core network to monitor traffic at packet level, reconstruct flows and classify them with a SVM. We want to support data rates going up to tens of Gb/sec as equipments such as NetFPGA 10G [28] or COMBOv2 [29] are available to test algorithms at this speed.

The goal for on-line traffic classification is to handle all flows on a saturated 10 Gb/s link. Two main functions will be required to achieve this goal:

- The flow reconstruction reads each packet, identifies to which flow it belongs, and stores the packet lengths required for classification. The processing speed depends on the number of packets per second in the traffic.
- The SVM classification runs the SVM algorithm once for each received flow. The processing speed depends on the number of flows per second in the traffic.

The traces used to test the classification algorithm are described in Table I. The average sizes of packets and flows in bytes vary for these traces. Requirements in terms of packets/sec. and flows/sec. supported by the algorithm to reach a 10 Gb/s speed are described for each trace in Table IV. To support each trace sent at 10 Gb/s, the flow reconstruction should support at least 55 861 124 packets/sec. and the SVM classifier should support at least 1 279 231 flows/sec.

We have first developed a software version of the classifier that is fed by a trace, to assess the possible performance in

Trace	Packets per second	Flows per second
Ericsson	6 809 840	27 805
Brescia	55 861 124	256 472
FT	48 310 718	1 279 231

TABLE IV
REQUIREMENTS FOR A CLASSIFICATION AT 10 GB/S FOR EACH TRACE

software. The classifier is made up of 3 main processes: (i) read the trace, (ii) rebuild flows from the stream of packets (iii) classify flows. For flow reconstruction, an algorithm proposed for a Netflow hardware implementation [30] is used. It has the advantage of requiring a constant time per packet and a bounded memory, which fits well with a hardware implementation. For the SVM algorithm, the libSVM [25] library (written in C) was chosen. To use all the cores of the processor, openMP [31] for libSVM is enabled.

Table V shows the performance of the software implementation on a 2.66 GHz 6-core Xeon X5650 with hyper-threading enabled and 12 GB of DDR3 RAM. It shows that the software implementation is not able to support 10 Gb/s. The best supported speed ranges from 2.32 Mb/s to 1597 Mb/s depending on the trace.

- The flow reconstruction speed does not depend on the trace as the flow reconstruction algorithm requires a constant time per packet. The only noticeable difference is for the biggest trace, FT, where the flow reconstruction probably suffers from the heavy CPU usage of the SVM classification.
- SVM classification is always more limiting than flow reconstruction (0.024 % of the requirements for 10 Gb/s in the worst case). Its speed depends on different factors including the number of support vectors in each SVM model: Brescia is the trace for which the learnt model has the most support vectors (24 758), then come FT (6 296) and Ericsson (4 341).

Trace	Packets per second (flow reconstruction)	Flows per second (classification)
Ericsson	5 189 293 76 % of 10Gb/s req.	4 655 17 % of 10Gb/s req.
Brescia	5 153 675 9.2 % of 10Gb/s req.	1 031 0.40 % of 10Gb/s req.
FT	4 336 677 9.0 % of 10Gb/s req.	311 0.024 % of 10Gb/s req.

TABLE V
PERFORMANCE OF THE SOFTWARE IMPLEMENTATION COMPARED TO 10GB/S REQUIREMENTS

Even with a powerful computer, a software implementation is not able to reach a 10 Gb/sec. speed, mainly due to its limited ability to parallelize the computation. This justifies the use of hardware acceleration. Different platforms may be used to provide hardware acceleration for network monitoring:

- Network processors are programmable in software and provide hardware-accelerated tools for tasks commonly required in network monitoring.
- Programmable cards with an integrated Field-Programmable Gate Array (FPGA) are very flexible and provide hardware access to the network interfaces.

To be able to explore fully the parallelism possibilities in the SVM classification algorithm, we have chosen to use a card with an FPGA that is more flexible than network processors. Two main vendors provide such cards: NetFPGA with the NetFPGA 10G card [28], which has 4 interfaces at 10 Gb/s, and INVEA TECH with the Combov2 [29] card, which has 2 to 4 interfaces at 10 Gb/s. Both cards integrate a Xilinx Virtex-5 XC5VTX240 FPGA. We are going to present an implementation of the flow reconstruction and SVM classification on this FPGA.

For flow reconstruction we use the same principles as [30] which considers an FPGA implementation of Netflow. SVM implementations on FPGA have also been proposed, but they are either focused on the learning phase [32] or not adapted to our classification algorithm [33] as they are restricted to two-class problems or using different kernels.

B. The SVM classification algorithm

The classification part of the SVM algorithm takes a vector as input and returns the class of that vector as an output. It works with few steps, repeated for each support vector. Algorithm 1 describes these steps. It is the multi-class implementation of the decision making procedure described in Section II-A. This pseudo-code has been written in order to enlight the possibilities to parallelize the algorithm.

Algorithm 1 SVM classification algorithm

$x \leftarrow$ the vector to classify
for all support vector x_i **do** {Main loop}
  $c_i \leftarrow$ the class of x_i
  $k_i \leftarrow K(x_i, x)$
  for all class $c_j \neq c_i$ **do** {Sum loop}
    $d \leftarrow$ index of the decision between c_i and c_j
    $S_d \leftarrow S_d + y_{d,i} \times \alpha_{d,i} \times k_i$
  end for
end for
for all decision d between c_i and c_j **do** {Comparison loop}
  if $S_d - b_d > 0$ **then**
    Votes $V_i \leftarrow V_i + 1$
  else
    Votes $V_j \leftarrow V_j + 1$
  end if
end for
Select class $c_n \leftarrow$ class with the highest votes V_n

The support vectors and the y, α and b values are part of the SVM model. Compared to the notations used in Section II-A, index d is added to identify the binary decision problem considered for the model values.

C. Operations

The important operations of the classification algorithm are in the main loop, for each iteration:

- the kernel computation (Equation 2) requires 3 integer additions (one per vector component), 3 integer multiplications (to compute the squares), one multiplication by a floating-point constant, and one exponential computation.
- the sum computation requires one floating-point multiplication and one floating-point addition. It is run 7 times for the 8 classes defined in Table II.

D. Parallelism

The Main loop is where most of the computation time is spent. It iterates many times (from 4 341 to 24 758 support vectors for the different traces presented in Table I) and includes complicated operations (exponential, multiplications). But it can be easily parallelized as each iteration does not depend on others. The only shared data is in the additive S values. These values have to be duplicated so that iterations are computed in parallel. Then as S is additive, the duplicated values can be merged by summing them.

The Sum and Comparison loops have few iterations: one per class. In this article there are 8 classes defined in Table II, so the loops can be totally parallelized. The Sum loop iterations are totally independent, while the Comparison loop iterations share the votes counter, which is additive. So it can be duplicated and then merged.

All loops can be removed by using parallel processing except the main loop, that has too many iterations and would require more area than is available on the Virtex-5. But it is possible to implement more than once the Main loop, so that less iterations are required to process one vector. Section IV-B describes an architecture with an adjustable level of duplication of this loop.

IV. Hardware implementation

A. Fixed-point model

Floating-point operations are complex to realize in hardware and use too much area on the FPGA. The best solution is to transform the algorithm to use a fixed-point model instead of a floating-point model. Table VI shows the bit widths of different variables used in the SVM fixed-point model.

Variable	Integer part	Decimal part
Vector component	11	0
α	7	11
γ	0	18
b	15	11
S	15	11

TABLE VI
QUANTIZATION OF THE MAIN SVM FIXED-POINT MODEL VALUES

These quantization parameters have been chosen so that the mathematical operations are made on values as small as possible, without loosing too much precision for the classification. Some sizes are quite large because the classifier should work whatever the SVM model, so that a new synthesis is not required to change the model. The possible values of the variables have been determined by analyzing SVM models learnt in different conditions. For example the precision of the γ parameter is very important (decreasing it leads to a drop in classification accuracy), but its absolute value never reaches 1. The 11-bit width of a vector component has been chosen because we assume that the size of a packet will not be more than 1500 bytes.

Multiplications are complex to realize in hardware. They are required to compute the squares in the kernel, but squares are symmetric functions with one integer parameter varying from -1500 to 1500. A ROM with 1501 values is used to emulate squares. Similarly, a ROM is used to emulate the exponential function. Finally, to avoid the $y_{d,i} \times \alpha_{d,i} \times k_i$ multiplication, $ln\left(|y_{d,i} \times \alpha_{d,i}|\right)$ is precomputed, and the exponential used to compute k_i is computed only after the addition of this term. Delaying the exponential computation transforms the multiplication into an addition. This way only one multiplication by a constant remains in the kernel computation, which is much simpler than a multiplication of two variables.

To check that the loss in precision is not too important, a software implementation of the classification algorithm with the fixed-point model has been implemented. Figure 2 compares the accuracy of the fixed-point model to the results of the float model as described in Section II-B. It shows that the transition to fixed-point decreases the accuracy of the algorithm, but it remains around 90 %. Depending on the requirements, a higher accuracy can be achieved using wider fixed-point values, but it will require more space on the FPGA.

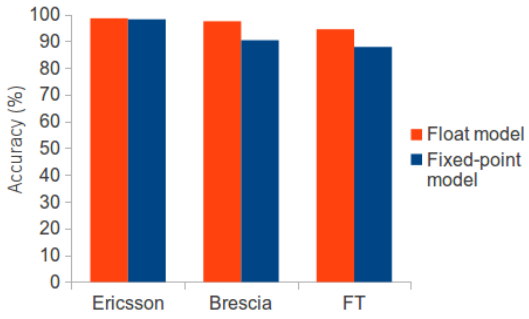


Fig. 2. Accuracy of the fixed-point model compared to the float model

B. Architecture

The architecture of a traffic-processing module on NetFPGA or Combov2 cards is very similar. It uses a block with an input bus for input traffic, and an output bus for output traffic. The classifier block is described in Figure 3.

The computation units represent the most important part of this architecture: they implement the computation of the main loop described in Algorithm 1. To get the best performance from the FPGA, operations of the algorithm must be parallelized. As seen in Section III-D, all loops can be totally unrolled by duplicating the hardware for each iteration except

Trace	Ericsson			Brescia		FT		
Computation units	2	4	8	2	8	2	4	8
Occupied slices	8 414	14 186	26 350	24 340	32 679	10 174	15 643	26 846
Occupied slice registers	9 221	21 864	45 967	9 272	40 658	8 966	21 287	44 356
FPGA usage (% of slices)	22.47	37.89	70.38	65.01	87.28	27.17	41.78	71.70
Maximum frequency (MHz)	174	156	165	51	62	157	164	139
Cycles per flow	2 193	1 110	569	12 401	3 121	3 170	1 598	813
Flows per second	79 733.6	140 766	290 107	4 122.42	20 164.3	49 741.0	102 840	171 861
% of 10Gb/s requirements	286.8	506.3	1043	1.61	7.862	3.888	8.039	13.43

TABLE VII
SYNTHESIS RESULTS OF SVM TRAFFIC CLASSIFICATION ON A VIRTEX-5 XC5VTX240 FPGA

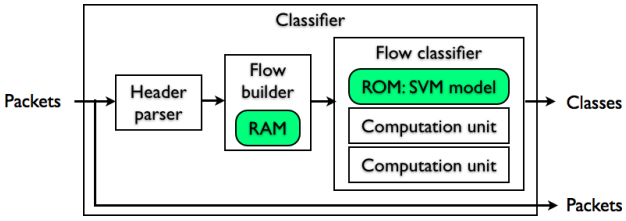


Fig. 3. Architecture of the classifier

the main loop. The computation unit is duplicated as much as the Virtex-5 supports.

As the computation in the main loop is complicated, each iteration will take many clock cycles in hardware. To improve the throughput of the loop and reduce its computation time, the iterations can be pipelined: one new support vector is processed by the first operation of the computation unit at each clock cycle, and then forwarded to the next operation. This way all operations work in parallel and each computation unit accepts one support vector at each time step.

As Figure 3 shows, the SVM model is currently stored in ROMs. This forces to synthesize the design again to change the SVM model used. In future works, ROMs will be converted into RAMs, so that it is possible to change the SVM model faster. This will not require to change the classifier implementation.

C. Results of the hardware accelerated traffic classifier

The proper behavior of the hardware implementation has been tested by checking that its results are exactly identical to the software implementation of the fixed-point model, first in simulation, and then implemented on a NetFPGA card.

To assess the performance of the hardware implementation and compare it to the software implementation, it has been synthesized on a Virtex-5 XC5VTX240. Three different SVM models (one for each trace) have been tested. The number of processing units has been changed as well, to exploit the maximum parallelism on the FPGA. Table VII presents the results of these synthesis.

The number of occupied slices and slice registers as well as the maximum frequency are given by the synthesis tool. They are an indication of the hardware complexity of the implementation. The number of cycles required per flow has been determined by analyzing the code of the hardware implementation. It increases with the number of support vectors in the model, and decreases with the number of parallel computation units.

Thanks to massive parallelism, hardware implementations all have better performance in terms of flows per second than software implementations. The implementation for the Brescia trace gives poor results because of its low working frequency. The particularity of this trace is that the SVM model contains more support vectors than the others. They use too much space on the FPGA, which creates long and slow routes in the design an decrease its maximum frequency. The Ericsson and FT traces SVM models have less support vectors. Even with only 2 computation units, the implementation for the Ericsson trace gives results much higher than the requirements to support a 10 Gb/s speed (286 % of the requirements). The implementation for the FT trace brings roughly the same performance improvements, but the requirements in terms of flows per second are very high because the trace contains many very small flows. So with 8 computation units, it fulfills only 13 % of the requirements.

A performance result that is not visible in Table VII is the delay that the classifier adds to the packets if it is used directly on a network link to tag packets with their class number (and not just as a passive probe set up in derivation on the link). The current implementation sends the classification to the host computer instead of tagging the packets, but it could be modified without overhead. For now, 10 clock cycles are required between the arrival of the packet in the classifier and the time when the class of the packet is known. This delay is constant because a packet is considered unknown if the flow has not yet been classified (its class is not in RAM), so it does not depend on the classification time. At the frequencies obtained in Table VII, the delay ranges from 57.2 ns to 254 ns. These figures do not include the time required to handle the Ethernet protocol.

To improve the supported speed for all traces, many directions are possible. A better management of the memory used to store the SVM models, by reducing it or putting it in an external memory, would bring more room to parallelize the logic on the FPGA. Critical paths in the design may be improved to achieve higher frequencies. It is also possible to use more powerful FPGAs, or to use multiple FPGAs in parallel to reach the 10 Gb/s speed for each trace. Algorithmic

changes to reduce the number of support vectors in the models may help too.

V. Conclusion and Future Works

In this paper, we have studied the practical implementation of SVM-based traffic classification. Our contribution concerns the performance of a hardware accelerated version of the classifier. We have adapted the classifier to high rate traffic with online hardware acceleration of the SVM detection algorithm on a FPGA. The first tests have demonstrated that our hardware implementation outperforms tremendously the software implementation and allows real-time classification. As a future direction of research, we will enhance our FPGA implementation to support higher speeds and dynamic classification models and test it within a production network. Globally, this work is a step further to the real implementation of a classification architecture.

Acknowledgment

This work has been partly funded by the European project DEMONS: DEcentralized MONitoring for trustworthinesS.

References

[1] CISCO Systems, "Cisco Visual Networking Index: Forecast and Methodology, 2010-2015," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[2] M. M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 135–148.

[3] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated construction of application signatures," in *In SIGCOMM 2005 MineNet Workshop*, 2005.

[4] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '05. New York, NY, USA: ACM, 2005, pp. 50–60.

[5] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 229–240, 2005.

[6] L. Bernaille, R. R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 23–26, 2006.

[7] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, ser. MineNet '06. New York, NY, USA: ACM, 2006, pp. 281–286.

[8] H. Dahmouni, S. Vaton, and D. Rossé, "A markovian signature-based approach to IP traffic classification," in *MineNet 2007 : ACM Sigmetrics Workshop on Mining Network Data*, 2007, pp. 29 – 34.

[9] T. Auld, A. Moore, and S. Gull, "Bayesian neural networks for internet traffic classification," *Neural Networks, IEEE Transactions on*, vol. 18, no. 1, pp. 223 –239, 2007.

[10] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 5–16, 2007.

[11] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and K. Claffy, "GT: picking up the truth from the ground for Internet traffic," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 13–18, 2009.

[12] A. Este, F. Gringoli, and L. Salgarelli, "On the stability of the information carried by traffic flow features at the packet level," *SIGCOMM Comput. Commun. Rev.*, vol. 39, June 2009.

[13] W. Li, M. Canini, A. Moore, and R. Bolla, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema," *Computer Networks, Special Issue on Traffic classification and its applications to modern networks*, vol. 53, pp. 790–809, 2009.

[14] C. Rotsos, J. J. Van Gael, A. Moore, and Z. Ghahramani, "Probabilistic graphical models for semi-supervised traffic classification," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 752–757.

[15] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification of P2P traffic," *Elsevier Computer Networks*, vol. 55, no. 6, pp. 1394–1411, 2011.

[16] H. Kim, D. Barman, M. Faloutsos, M. Fomenkov, and K. Lee, "Internet traffic classification demystified: The myths, caveats and best practices," in *In Proc. ACM CoNEXT*, 2008.

[17] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[18] S. Valenti, *Dealing with P2P traffics in modern networks: measurement, identification and control*. Paris: TELECOM ParisTech, 2011.

[19] D. Rossi and S. Valenti, "Fine-grained traffic classification with netflow data," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10, 2010, pp. 479–483.

[20] N. Williams, S. Zander, and G. Armitage., "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Computer Communication Review*, 2006.

[21] A. Este and F. Gringoli, "On-line svm traffic classification," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '11, 2011.

[22] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[23] G.Gomez and P.Belzarena, "Early Traffic Classification using Support Vector Machines," in *Fifth International Latin American Networking Conference*, ser. LANC'09, 2009.

[24] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for tcp traffic classification," *Computer Networks*, vol. 53, no. 14, pp. 2476 – 2490, 2009.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[26] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

[27] Clear Foundation, "l7-filter: application layer packet classifier for Linux," http://l7-filter.clearfoundation.com/.

[28] NetFPGA, "NetFPGA: a line-rate, flexible, and open platform for research, and classroom experimentation," http://netfpga.org/.

[29] CESNET, "Our hardware," http://www.liberouter.org/hardware.php?flag=U, Nov. 2011.

[30] M. Žádník and L. Lhotka, "Hardware-accelerated netflow probe," Technical Report 32/2005, CESNET, Praha, Tech. Rep., 2005.

[31] L. Dagum and R. Menon, "Openmp: an industry standard api for shared-memory programming," *Computational Science Engineering, IEEE*, vol. 5, no. 1, pp. 46 –55, jan-mar 1998.

[32] D. Anguita, A. Boni, and S. Ridella, "A digital architecture for support vector machines: theory, algorithm, and fpga implementation," *Neural Networks, IEEE Transactions on*, vol. 14, no. 5, pp. 993 – 1009, sept. 2003.

[33] D. Anguita, S. Pischiutta, S. Ridella, and D. Sterpi, "Feed-forward support vector machine without multipliers," *Neural Networks, IEEE Transactions on*, vol. 17, no. 5, pp. 1328 –1331, sept. 2006.

DETECTION D'ANOMALIES DANS LE TRAFIC

LES SYSTEMES DE DETECTION D'INTRUSIONS : ETAT DE L'ART ET ENJEUX

Attaques contre les réseaux informatiques

Plusieurs raisons font qu'on assiste à une prolifération des problèmes de sécurité informatique et réseaux depuis le début des années 2000. L'essor de l'Internet et le développement du haut débit ont multiplié le nombre d'organisatons potentiellement impliquées dans des cyber-attaques. La multiplication des terminaux hétérogènes, le développement et l'essor des nouveaux services, le développement des réseaux overlay (P2P) et des technologies d'accès sans fil ont introduit de nouvelles sources de vulnérabilités. La sophistication et la sévérité des cyber-attaques a augmenté considérablement en une dizaine d'années.

Des organisations mafieuses ont investi le domaine des systèmes d'information et des réseaux. Les systèmes d'informations sont devenus des organes vitaux pour beaucoup d'organisations (compagnies, gouvernements, etc...) et les réseaux de télécommunications jouent le rôle de système nerveux central de notre société. Les systèmes d'information et les réseaux sont donc des cibles privilégiées pour qui veut nuire à une organisation. Il est devenu possible d'acheter les services de réseaux de malfaiteurs pour orchestrer une attaque contre une cible particulière. Des outils sont disponibles sur le Web un peu partout dans le monde ce qui facilite la tâche à certains individus mal intentionnés.

Les cyber-attaques sont souvent médiatisées. Les cibles de ces attaques peuvent être par exemple des sites de E-commerces (Ebay, Amazon, etc...), des réseaux sociaux (Twitter, Facebook, août 2009), des banques, des organisations gouvernementales. Les attaques peuvent viser des serveurs Web (CNN, Yahoo, Ebay, Amazon, DoubleClick, Authorize.net), des serveurs DNS racine, etc....

L'actualité récente fournit de nombreux exemples d'attaques réussies :

- Janvier 2012 : attaques par le groupe de hackers Anonymous contre diverses infrastructures suite à la fermeture du site MegaUpload par le FBI
- Noël 2011 : attaque de DDoS contre le site du Sénat. L'attaque est liée à celle qui a visé le site de la députée UMP Valérie Boyer, auteur de la proposition de loi réprimant la négation du génocide arménien. L'attaque contre le site du Sénat est revendiquée par le hacker Iskorpitx, et celle contre le site de Valérie Boyer par Grayhatz, une communauté de hackers turcs.
- Novembre 2011 : attaque informatique contre le site de Charlie Hebdo conjointement à l'incendie du siège du magazine suite à la publication d'une édition spéciale "Charia Hebdo" en réaction à la victoire du parti islamiste Ennahdha lors des récentes élections tunisiennes ; le piratage du site est revendiqué par un certain Akincilar, groupe de hackers turcs qui officient sur Cyber-Warrior.org.
- Septembre 2011 : Areva, spécialiste français du nucléaire, est victime d'une intrusion sur un de ses réseaux informatiques déclaré comme non critique par la société ; l'attaque vise le vol d'informations sensibles et perdurait probablement depuis un certain temps au moment de sa découverte.
- Mars 2011 : attaque contre l'autorité de certification Comodo : une intrusion sur le site de l'autorité de certification a permis de générer frauduleusement 9 certificats concernant 7 domaines (sites de messagerie : Yahoo Mail, Hotmail et Gmail ; Skype ; plate-forme de téléchargement d'extensions de Mozilla et Global Trustee). La détention de tels certificats permet de leurrer les internautes et de réaliser des attaques de type phishing en usurpant l'identité d'un site légitime pour voler des informations personnelles (mots de passe, etc...). Les attaques provenaient d'adresses IP iraniennes. La motivation semblait être politique, la surveillance de ces sites de communication pouvant être d'une grande utilité pour des gouvernements cherchant à surveiller les communications d'un groupe de dissidents. L'attaque a eu lieu en mars 2011 au moment du "Printemps Arabe".
- Fin 2010 : attaques de DDoS contre le site de WeakiLeaks qui publie des mémos diplomatiques ; contre-attaques de pro-WeakiLeaks contre différents sites ...

Les attaques contre un système d'information peuvent être de différents types. Les attaques de Déni de Service (DoS, Denial of Service) consistent à priver d'accès les utilisateurs autorisés en inondant le système (par ex. un ou des serveurs) de demandes que le système ne peut satisfaire et qui consomment toutes ses ressources. Les activités de type scanning/probing ont pour objectif de découvrir des informations sur le système, par exemple quels sont les services offerts ou quel est le système d'exploitation. Ces activités de découverte du système sont souvent une étape préalable à une attaque. Les vers sont des programmes informatiques qui s'auto-répliquent, se multiplient et se propagent d'une manière furtive dans le système d'information. Les chevaux de Troye sont des programmes d'apparence légitime utilisés pour ouvrir une porte dérobée permettant à un attaquant d'obtenir un accès sur une machine. Les machines peuvent être compromises soit parce qu'un accès a été obtenu par un utilisateur non autorisé soit parce qu'un utilisateur a réussi à outre-passer ses droits pour exécuter des opérations privilégiées. Les machines compromises peuvent faire partie d'un réseau de machines zombies (botnet) contrôlé par une machine maître qui utilise les zombies pour lancer de manière distribuée des attaques contre une machine cible.

Taxonomie des systèmes de détection d'intrusions

On appelle intrusions des activités qui ont pour objectif d'outre-passer les mécanismes de sécurité des systèmes informatiques, par exemple l'accès au système d'un utilisateur non autorisé au travers de l'Internet ou un utilisateur autorisé outre-passant ses droits sur une machine. Les pare-feux (en anglais, *firewalls*) ont pour objectif de faire respecter la politique de sécurité d'un réseau, celle-ci définissant les communications autorisées sur un réseau informatique et en particulier la politique d'accès aux ressources de type serveurs. Le principe des pare-feux est de filtrer le trafic sur la base de différents critères (adresses IP, ports, motif dans la charge utile des paquets, etc...)

Cependant comme le prouvent les nombreuses attaques récentes réussies et médiatisées les pare-feux ne suffisent pas à stopper toutes les intrusions. Ils perdent de leur importance du fait du développement des communications chiffrées (HTTP sur SSL) qui court-circuitent tout filtrage. Par ailleurs les attaquants sont parfois déjà à l'intérieur du réseau et donc hors de tout contrôle du pare-feu. Les systèmes de détection d'intrusion (IDS, Intrusion Detection Systems) complètent l'arsenal de sécurité d'un système d'information. Le rôle d'un IDS est d'analyser les événements qui ont lieu sur un système d'information de façon à repérer des activités anormales ou suspectes (sur un réseau ou sur un hôte). Ils permettent d'avoir connaissance des tentatives échouées ou réussies d'intrusions et de déclencher une alarme en cas de suspicion d'intrusion. On parle ici d'intrusion au sens large c'est-à-dire toute activité ayant pour objet d'outre-passer les mécanismes de sécurité du système pour attenter à sa confidentialité, son intégrité, sa disponibilité, etc.

Les systèmes de détection d'intrusion peuvent être classés en IDS basés hôte (*Host based IDS*, HIDS) ou en IDS basés réseau (*Network based IDS*, NIDS). Les HIDS, basés hôte, analysent des événements obtenus à partir d'un seul hôte, comme des appels systèmes, des données de log, etc... Les NIDS, basés réseau, obtiennent des informations en analysant le trafic du réseau auquel les machines sont connectées. Par ailleurs certains IDS analysent les alarmes remontées par d'autres IDS de niveau plus bas pour les corréler et fournir une analyse de plus haut niveau.

Traditionellement les systèmes de détection d'intrusions (Snort, Bro, etc...) sont basés sur des signatures d'attaques connues. Des filtres sont définis sur la base des signatures connues de certaines attaques. On associe à chacun de ces filtres une règle qui spécifie comment l'IDS doit réagir (par exemple génération d'un message d'alerte) dans le cas où le trafic analysé est conforme au filtre. Une limitation des IDS basés signature est qu'ils ne peuvent pas reconnaître de nouvelles attaques, pour lesquelles une signature n'a pas encore été intégrée dans la base des règles de l'IDS.

Les IDS basés anomalies ne souffent pas de cette limitation. Le principe des IDS basés anomalies est de se baser sur un modèle (statistique) représentant le trafic indemne de toute anomalie. Une fois ce modèle défini l'IDS génère une alerte dans le cas où le trafic analysé s'écarte du manière statistiquement significative du modèle de trafic normal. La limitation des IDS basés signature est leur incapacité à détecter des attaques nouvelles (*zero day*) et donc leur tendance à manquer des attaques. De plus les IDS basés signature ne sont pas capables de détecter des attaques quand le trafic est chiffré et sont mis en difficulté par les attaques polymorphiques. La limitation des IDS basés anomalie est au contraire leur tendance à générer beaucoup de fausses alarmes et la nécessité de disposer d'un ensemble d'apprentissage représentatif et indemne de toute attaque pour calibrer le modèle de trafic "sain".

Systèmes de détection d'intrusion basés anomalies

Les IDS basés anomalies fondent leur décision sur des outils de fouille de données. Un modèle du trafic sain est construit par analyse de jeux de données supposés sans anomalies. Les anomalies sont détectées comme une déviation significative par rapport au modèle de trafic sain. Les IDS basés anomalies utilisent des méthodes d'apprentissage automatique. Différentes méthodes sont utilisées dans la littérature. On peut citer par exemple l'analyse en composantes principales (PCA, *Principal Component Analysis*), des décompositions en ondelettes ou décomposition en modes empiriques (EMD, *Empirical Mode Decomposition*), l'algorithme EM (EM, *Expectation Maximisation*), les machines à vecteurs de support (SVM, *Support Vector Machines*) ...

Les performances des algorithmes se mesurent en termes de probabilité de fausse alarme (FAR, *False Alarm Rate*) et de puissance. Le FAR mesure la probabilité qu'une alarme soit levée alors que le trafic est sain. La puissance du test mesure la probabilité de générer, à juste titre, une alarme en présence d'une anomalie. Il existe en général un compromis entre la puissance et le FAR. Le réglage de la sensibilité du détecteur se fait au travers du choix d'une valeur seuil au-delà de laquelle l'alarme est générée. Plus le seuil est élevé moins le détecteur générera de fausses alarmes mais également moins il sera sensible aux anomalies. Ce compromis entre FAR et puissance est souvent visualisé sous la forme d'une courbe ROC (ROC, *Receiver Operating Characteristic*) qui représente les valeurs du couple (FAR, puissance). Chaque point de cette courbe orrespond à une valeur particulière de seuil. Lors de la conception d'un IDS on souhaite une courbe ROC qui se rapproche du fonctionnement idéal c'est-à-dire du point de fonctionnement (FAR = 0, puissance = 1).

Validation des performances sur jeux de données

Il est possible de comparer les performances de différents IDS en analysant les résultats qu'ils fournissent sur des jeux de données de test. Un effort a été entrepris pour mettre à disposition de la communauté académique des jeux de données avec identification des anomalies pour servir de *benchmark*. Le jeu de données public le plus connu dans le domaine de la détection d'intrusion est certainement le jeu de données DARPA [15]. Le jeu de données DARPA 1998 et sa modification KDDCup99 ont été créés dans le cadre du projet MADAM ID. Les jeux de données DARPA 1999, DARPA 1998 et KDDCup99 sont aujourd'hui considérés comme peu représentatifs car trop anciens. Il existe également des traces au niveau paquets comme celles de l'université de Melbourne : les jeux de données Auckland II et MOAT. L'analyse peut également se faire au niveau des appels systèmes auquel cas un jeu de données comme celui de l'université du Nouveau Mexique peut être utilisé [19]. L'observatoire du réseau Abilène (Internet 2 Observatory, [11]) fournit également une source de données utile.

Les chercheurs génèrent bien souvent leur propre jeu de données en lançant eux-mêmes différents comportements malicieux et en enregistrant simultanément les mesures faites par différents capteurs (logs des appels systèmes, capture du trafic au niveau paquet, etc...) comme nous l'avons fait dans le cadre du projet OSCAR [1] financé par l'Agence Nationale de la Recherche (ANR). Dans ce cas-là il est possible d'évaluer d'une manière fiable la capacité de l'IDS à détecter les comportements malicieux mais il est souvent difficile de mesurer le taux de fausse alarme à moins d'être dans un environnement entièrement contrôlé et isolé de l'Internet.

Enjeu du haut débit dans le domaine de la détection d'anomalies de trafic

La conception des IDS est confrontée à un certain nombre d'enjeux. Le premier enjeu réside dans la masse de données à analyser et le fait que les anomalies sont par nature des événements rares. L'IDS basé anomalies est donc à la recherche d'une "aiguille dans une botte de foin". Cela pose deux problèmes distincts. D'une part il est difficile de détecter un événement rare de manière fiable sans générer de nombreuses fausses alarmes. D'autre part la masse des données à analyser est énorme et de plus, dans certains cas, cette analyse doit être faite en temps réel.

Il est nécessaire d'être capable d'analyser de très grosses masses de données. Il est par ailleurs nécessaire d'analyser le trafic à des débits réalistes, allant de quelques centaines de Mb/sec au niveau de l'accès à la dizaine de Gb/sec si on considère une analyse en coeur de réseau. Cette analyse doit si possible être

faite en temps réel c'est-à-dire que l'analyse de 1 seconde de trafic doit prendre moins de 1 seconde de traitement. La réactivité du système conditionne la possibilité de mettre en place des contre-mesures pour limiter l'impact de l'attaque.

Un sous-échantillonage des données à analyser peut constituer une solution partielle au problème du volume de données à traiter à la manière du sous-échantillonage dans les mesures NetFlow. Cependant ce sous-échantillonnage peut masquer certaines anomalies. Les contraintes de débit et de grosses masses de données à analyser constituent donc un challenge à relever pour la recherche dans le domaine des IDS.

Plusieurs approches doivent être combinées : des approches algorithmiques pour l'analyse de très grosses masses de données, éventuellement de manière séquentielle (*data mining*, *data stream mining*), accélération logicielle par parallélisation de code, accélération matérielle avec conception de circuits dédiés...

Alors que la conception de méthodes algorithmiques pour les IDS qui cherchent à détecter des anomalies dans le trafic a donné lieu à une littérature très prolifique, la question de leur implémentation pratique n'a pratiquement pas été abordée. La recherche académique sur les IDS basés anomalie semble ne pas avoir conduit à un déploiement à grande échelle dans le domaine commercial [17]. Il existe cependant quelques produits implémentant une approche basée anomalie pour la détection d'intrusions, par exemple IDES/NIDES, EMERALD, SPADE, Computer Watch (AT&T), Wisdom & Sense, ...

Enjeu des attaques distribuées et inter-domaines dans le domaine de la détection d'anomalies de trafic

Un autre enjeu est lié au caractère de plus en plus sophistiqué des attaques et à leur nature distribuée. Il est nécessaire de concevoir des systèmes de détection d'intrusion flexibles et réactifs, et capables de passer à l'échelle pour analyser du trafic à haut débit. Les attaques sont la plupart du temps distribuées. Bien souvent une machine prend le contrôle d'un certain nombre d'autres machines et constitue un réseau de machines compromises sous son contrôle (*botnet*). Les machines du *botnet* sont utilisées pour lancer une attaque distribuée vers une victime. Le caractère distribué de l'attaque contribue à sa furtivité. En effet les événements associés à une seule machine zombie peuvent être d'apparence anodine, alors que leur effet cumulé sur l'ensemble des machines participant au *botnet* constitue une attaque sévère contre la machine cible.

Le caractère distribué des attaques rend leur détection plus difficile car pour les détecter il faut être capable de corréler des événements relatifs à un grand nombre de machines zombies. Il est nécessaire de mettre en place des sondes en différents points du réseau et d'agréger de manière intelligente les informations remontées par ces différentes sondes. Par ailleurs les machines zombies peuvent se trouver sur des réseaux sous le contrôle d'entités administratives différentes ce qui complique considérablement la corrélation car elle nécessite un échange d'informations entre ces entités administratives.

Les victimes sont parfois localisées un peu partout dans le monde ; certaines attaques se sont propagées à une vitesse impressionnante et avec une portée mondiale. Les attaques les plus impressionnantes sont donc des attaques inter-domaines. Le réseau Internet est constitué de milliers de systèmes autonomes (*Autonomous Systems*, AS) interconnectés entre eux pour assurer une connectivité de l'Internet au niveau mondial. Les systèmes autonomes échangent du trafic entre eux selon différents accords, des accords de *peering* dans le cas de réseaux d'importance similaire ou des accords client/fournisseur, les réseaux les plus petits rétribuant les réseaux plus importants pour transporter leur trafic.

L'Internet est un écosystème compliqué, les systèmes autonomes étant contraints de coopérer pour construire un Internet mondial mais étant également en compétition les uns avec les autres [6]. Du fait de cette compétition les AS sont particulièrement soucieux de préserver leur confidentialité et souhaitent d'une manière générale communiquer aussi peu d'informations que possible sur leur topologie interne, leur trafic, la QoS offerte par leur réseau, etc...

Pour contrer de manière efficace les cyber-menaces au niveau mondial il est souhaitable de mettre en place une collaboration entre les centres de surveillance en charge des différents domaines. La communication d'informations (remontées d'alertes, identification de machines suspectes, etc...) entre les différents domaines est donc souhaitable pour augmenter le niveau de sécurité du réseau. Mais ce besoin de communication est en contradiction avec le souci de confidentialité des différents domaines et peut également être en opposition avec les différents cadres législatifs nationaux, concernant en particulier le respect de la vie privée. A notre connaissance la communication d'alertes entre les centres de surveillance n'est pas automatisée ; elle se fait essentiellement sur la base de relations de confiance entre les individus, d'appels

téléphoniques...

Il est nécessaire de construire un cadre permettant une meilleure coordination de la surveillance des réseaux afin de permettre une détection collaborative des cyber-menaces et la mise en place concertée de contre-mesures. Il est nécessaire d'automatiser cette coordination car certaines attaques se propagent très rapidement et atteignent de nombreuses machines en quelques minutes seulement. Le sujet est particulièrement complexe tant du point de vue technique que du point de vue des enjeux des différents acteurs impliqués et des contraintes imposées par les différents cadres législatifs nationaux. C'est le challenge que doit relever le projet intégré du 7ème programme cadre de recherche (PCRD 7) financé par la Commission Européenne *DEMONS : DEcentralized MONitoring for trustworthinesS* (septembre 2010- mars 2013, [9]) auquel nous participons.

TROIS CONTRIBUTIONS

CALLEGARI Christian, VATON Sandrine, PAGANO Michele, A New Statistical Method for Detecting Network Anomalies in TCP Traffic, European Transactions on Telecommunications (ETT), volume 21, issue 7, november 2010

Cet article porte sur le développement d'un IDS basé anomalie qui analyse les connexions TCP au niveau d'un hôte (HIDS basé anomalie). Différentes attaques peuvent exploiter le comportement des connexions TCP, l'attaque la plus connue étant le TCP SYN flooding. Cette attaque exploite la phase de négociation initiale (*handshake*) des connexions TCP. Elle vise à atteindre un Déni de Service (DoS) destiné à faire tomber un serveur en lui envoyant de nombreux paquets SYN sans répondre aux accusés de réception SYN-ACK ce qui oblige le serveur à maintenir un état pour chaque connexion semi-ouverte au détriment des demandes de connexion légitimes.

Il est d'une manière générale intéressant d'étudier la détection de connexions TCP mal formées. L'approche comportementale consiste à se baser sur un modèle statistique des connexions TCP ; les connexions qui s'écartent sensiblement de ce modèle de comportement moyen sont identifiées comme des connexions "suspectes".

TCP permet un dialogue entre le client et le serveur ; ce dialogue a pour but de détecter les pertes de paquets afin de les retransmettre, d'adapter le débit de la connexion aux conditions réseaux de façon à réaliser un partage harmonieux des ressources, de signaler des paquets à transmettre prioritairement à la couche applicative... Ce dialogue entre client et serveur se fait au travers de différents champs de l'entête, en particulier les drapeaux TCP (6 bits SYN, ACK, PSH, RST, URG, FIN).

Ces 6 bits peuvent être identifiés à un entier compris entre 0 et 63, la taille de l'espace des états est en réalité beaucoup plus petite que 64 puisque certaines combinaisons de drapeaux ne se produisent jamais. Estevez et al. [8] ont proposé de modéliser les valeurs des drapeaux des paquets successifs d'une connexion TCP comme une chaîne de Markov à temps discret et à état discret et de baser un IDS sur cette modélisation stochastique des drapeaux TCP dans le protocole.

Par ailleurs Ju et al. [13] ont proposé d'utiliser un modèle de chaînes de Markov *d'ordre élevé* pour modéliser le comportement des utilisateurs d'un hôte et détecter les mascarades. Une mascarade est une tentative pour des utilisateurs non autorisés de se faire passer pour un utilisateur autorisé. La suite des commandes saisies par chaque utilisateur autorisé est modélisée par une chaîne de Markov d'ordre élevé ; un test basé sur la valeur de vraisemblance permet de détecter qu'une nouvelle séquence de commandes ne correspond pas au profil des utilisateurs autorisés du système.

Nous avons souhaité évaluer l'intérêt de l'utilisation des modèles de chaîne de Markov d'ordre élevé pour la conception d'un IDS basé anomalie. L'estimation des paramètres d'une chaîne de Markov d'ordre élevé (probabilités de transition et distribution initiale) à partir d'un ensemble d'apprentissage est un problème intéressant. L'ensemble d'apprentissage est dans notre cas constitué d'un grand nombre de connexions TCP supposées être dénuées d'anomalies.

Dans le cas des chaînes de Markov d'ordre 1, les paramètres peuvent être estimés très simplement en comptant le nombre de transitions entre états de la chaîne de Markov dans l'ensemble d'apprentissage. Par contre dans le cas des chaînes de Markov d'ordre élevé l'estimation des paramètres du modèle est un

problème complexe. Pour résoudre le problème nous utilisons l'approche proposée dans [13]. Le nombre de paramètres à estimer pour un modèle de chaîne de Markov d'ordre élevé est très grand ; pour se ramener à une paramétrisation parcimonieuse du modèle (petit nombre de paramètres) on utilise un modèle MTD (*Mixture Transition Distribution*). Les paramètres du modèle sont estimés au sens du Maximum de Vraisemblance d'une manière itérative en alternant deux étapes, une étape utilisant de la programmation quadratique séquentielle et l'autre étape résolvant un problème linéaire inverse avec contraintes de positivité (LININPOS, *Linear Inverse Problem with Positivity Constraints*). Le problème LININPOS est résolu en utilisant un algorithme EM (*Expectation Maximization*) [22].

Les performances de l'IDS proposé sont testées sur deux jeux de données : le jeu de données DARPA-99 [15] et un jeu de données collectées sur le réseau du département d'ingénierie de l'information de l'université de Pise (environ 30 machines sous Windows, Linux et Mac OS) [20] [21]. Les performances d'un IDS basé sur une chaîne de Markov d'ordre 1 et d'un IDS basé sur une chaîne de Markov d'ordre plus élevé sont comparées.

Sur le jeu de données DARPA-99 le modèle d'ordre 4 donne de meilleures performances que le modèle d'ordre 1. Cependant l'amélioration n'est pas très significative car le modèle d'ordre 1 donne déjà d'excellents résultats avec un taux de reconnaissance des anomalies proche de 100% et un taux de fausse alarme proche de 0%. Pour modérer ces excellents résultats il est important de remarquer que DARPA-99 est un jeu de données de synthèse et que de plus l'apprentissage du modèle a été fait sur le même jeu de données que la détection des anomalies.

Les performances de l'IDS ont ensuite été évaluées sur du trafic de production mesuré sur le réseau du département d'ingénierie de l'information de l'université de Pise avec une injection volontaire de trafic malicieux. Le trafic malicieux correspondait à 115 attaques différentes de types variés (ICMP flooding, Teardrop, TCP flooding, craquage de mot de passe, scan) lancées en utilisant des outils Open Source.

Pour tester la stabilité de l'IDS nous avons décidé de réaliser l'apprentissage du modèle sur le jeu de données DARPA-99 et d'utiliser les modèles obtenus pour détecter les anomalies dans le jeu de données de l'université de Pise. Ce contexte est bien sûr défavorable car le jeu de données DARPA-99 est ancien alors que celui de l'université de Pise date de 2009 et contient du trafic multimédia (VoIP, Skype) et du trafic pair-à-pair (eMule/aMule, BitTorrent).

Les performances dans ce scénario plus réaliste restent très bonnes. Le modèle se comportant le mieux est le modèle d'ordre 3 puisqu'il permet de détecter 90% des attaques avec un taux de fausse alarme de 10%. Les modèles d'ordre 3 et 4 sont les seuls à permettre la détection de 100% des anomalies en augmentant le taux de fausse alarme.

Le gain en performance obtenu en utilisant des modèles de chaîne de Markov d'ordre élevé est à mettre en regard de la complexité de la calibration de ces modèles. Même si il est possible de développer des librairies pour faciliter l'usage de ces modèles, force est de constater que leur théorie est beaucoup plus complexe que celle des chaînes de Markov d'ordre 1.

SALEM Osman, VATON Sandrine, GRAVEY Annie, A scalable, efficient and informative approach for anomaly-based Intrusion Detection Systems : theory and practice, International Journal of Network Management (IJNM), Special Issue : Traffic Monitoring and Network Measurements : from Theory to Practice, volume 20, issue 5, september/october 2010

Cette publication fait état de travaux développés dans le cadre du projet de l'Agence Nationale de la Recherche *OSCAR : Overlay Security, Characterization, Analysis and Recovery* (2006-2008) [1]. Dans le cadre de ce projet nous nous sommes intéressés à la détection d'anomalies dans le trafic par un opérateur de réseau à partir de points de capture situés dans le réseau d'infrastructure.

Beaucoup d'approches en détection d'anomalies dans le trafic réseau agrègent tout le trafic mesuré au niveau paquet en une seule série temporelle et appliquent une méthode de détection de changement pour détecter l'occurrence d'une anomalie. Cette approche a de bonnes performances en termes de complexité et de passage à l'échelle mais comme tout le trafic est agrégé en une seule série temporelle, les anomalies de faible intensité passent inaperçues au milieu du reste du trafic, surtout si le point de capture est situé dans le coeur de réseau. Par ailleurs détecter l'instant d'occurrence d'une anomalie sans plus d'informations sur la nature de cette anomalie (adresses IP, type d'anomalie, etc...) est insuffisant pour permettre la

mise en place de contre-mesures en temps réel.

Dans ce contexte nous avons conçu une méthode d'analyse de trafic réseau permettant de découvrir des changements dans les volumes de trafic associés à une certaine clé. La clé considérée peut être par exemple une adresse IP source ou destination ou un numéro de port source ou destination ou encore une combinaison de ces champs d'entêtes. Une augmentation brusque du volume de trafic (par exemple nombre de paquets TCP SYN) associé à une clé particulière est une indication d'anomalies comme des attaques de flooding ou des activités de scan.

La valeur de la clé associée à une anomalie n'étant pas connue à l'avance il faut être capable de surveiller en permanence l'ensemble de l'espace des clés. Cet espace est en règle générale de très grande taille (2^{32} valeurs possibles pour une adresse IPv4 par exemple). Il est donc déconseillé de tenter de maintenir une liste de clés et des volumes de trafic associés car on atteint rapidement les limites en termes de mémoire. Les techniques de fouille de données en temps réel (*data stream mining*) permettent au contraire de retrouver des informations dans des flots de données prenant leurs valeurs dans des espaces de très grande dimension. En particulier il est possible en utilisant une structure de données appropriée appelée Count Min Sketch (CMS, [4]) de maintenir un résumé compact des volumes de trafic associés à l'ensemble de l'espace des clés.

Le principe de notre approche est de maintenir une structure de données appelée *sketch* et d'utiliser des méthodes de théorie de la décision séquentielle pour détecter avec un délai faible et un faible taux de fausse alarme une augmentation brusque dans la valeur de certaines cellules de cette structure de données.

L'algorithme proposé a été conçu de manière à permettre un traitement en temps réel du trafic. Il se compose de trois blocs principaux : (i) filtrage du trafic sur lequel on souhaite maintenir des statistiques (par exemple, filtrage des paquets TCP SYN) (ii) agregation du trafic filtré dans une structure de données de type *sketch* de façon à construire un résumé compact des volumes de trafic associés aux clés (par exemple un nombre de paquets TCP SYN correspondant aux différentes adresses IP destination) (iii) détection par un une méthode de théorie de la décision séquentielle (par exemple CUSUM ou des versions dérivées du CUSUM) de changements brusques dans les valeurs de certaines cellules de la structure de *sketch*. Un rappel sur le CMS [4] et sur le Multi-Layer Reversible Sketch peut être trouvé dans l'article. Un rappel sur certains algorithmes de théorie de la décision séquentielle comme le MultiChart Non Parametric CUSUM (MNP-CUSUM) [18] est également fait dans l'article.

Il est possible de retrouver la valeur de clé correspondant aux cellules détectées en analysant les indices de ces cellules. Ceci fournit une information détaillée permettant de mieux localiser l'anomalie, c'est-à-dire de déterminer les adresses IP ou numéros de port impliqués. Cette information détaillée est utile pour la mise en place de contre-mesures. De plus, pour permettre la détection et classification d'anomalies de différents types (DoS, DDoS, scan de port, scan de réseau, flashcrowd, etc...) nous maintenons en parallèle plusieurs instances du même algorithme. Chaque instance correspond à une définition différente de la clé utilisée pour indexer les cellules dans la structure de *sketch*. Un système expert combine les sorties des différentes instances de notre détecteur de façon à reconnaître différentes catégories d'anomalies.

La classification des anomalies exploite notre compréhension des signatures des différentes activités malicieuses. Par exemple une attaque de TCP SYN flooding peut être détectée comme une augmentation brusque du nombre de paquets TCP SYN vers une destination particulière. Une définition adaptée de la clé sera alors le couple (IP destination, port destination). Si l'attaque n'est pas distribuée on observera une augmentation brusque du nombre de paquets SYN associé à une adresse IP source particulière vers ce couple (IP destination, port destination). Si l'attaque est distribuée aucune adresse IP source ne sera détectée comme suspecte. Une activité de scan de port peut être détectée comme une augmentation brusque du nombre de paquets SYN vers une adresse IP destination particulière sans qu'aucun port correspondant à cette destination ne se distingue par une augmentation brusque du nombre de paquets SYN reçus. Au contraire une activité de scan réseau se caractérise par un grand nombre de paquets vers un numéro de port particulier sans qu'aucune adresse IP destination du sous-réseau considéré ne soit détectée comme destinataire d'un grand nombre de ces paquets.

Les performances du schéma proposé ont été testées d'une manière extensive. La validation de l'algorithme a été réalisée sur de nombreuses traces publiques (Abilene, Auckland-II, etc...) ainsi que sur des traces mises à disposition par France Télécom dans le cadre du projet OSCAR. Les traces fournies par France Télécom étaient de deux types : (i) une trace collectée sur une plaque ADSL d'Orange (ii) une trace collectée sur Open Transit (OTIP), le réseau de transit international de France Télécom. Par

ailleurs des expérimentations ont été menées entre les partenaires du projet pour tester les capacités de détection en temps réel des algorithmes.

Pour mettre en place les expériences de détection en temps réel de différents types d'anomalies (DoS, scan, etc...) une plateforme de mesure a été déployée entre les partenaires du projet. Des points de capture du trafic ont été mis en place dans les laboratoires de certains partenaires, dont Télécom Bretagne. Ces points de capture se basent sur une carte DAG [7] spécialisée dans la capture de trafic au niveau paquet avec une grande précision temporelle. Le trafic mesuré au niveau paquet grâce à la DAG était converti sur les machines hébergeant les DAG en des rapports de flots dans un format "OSCARFIX" défini par le projet. Les rapports de flot OSCARFIX étaient envoyés régulièrement par sockets UDP à un collecteur centralisé. Le collecteur centralisé récupérait les rapports de flots envoyés par les différents points de capture et exécutait divers algorithmes de détection, dont celui de Télécom Bretagne. Des attaques de différents types étaient lancées par France Télécom à des instants particuliers vers les machines participant à l'expérimentation. Des machines du réseau Planetlab ont été utilisées pour générer du trafic de charge et en particulier du trafic P2P.

Les résultats des différentes expérimentations sont décrits en détail dans l'article. L'algorithme a un très faible coût de calcul ; à titre d'exemple 3 jours de rapports NetFlow d'un routeur CISCO du réseau OTIP (896×10^5 flots, 6,9 GO de données) a été analysé en quelques minutes sur un Pentium à 2,2 GHz avec 3 GO de RAM. On démontre la capacité de l'algorithme à détecter des attaques de faible intensité dans le trafic de charge. Les expérimentations prouvent également sa capacité à déterminer le type d'attaques (par exemple un scan de port à la recherche d'un serveur SSH) ainsi qu'à déterminer les adresses IP des victimes (par exemple les victimes d'un SYN flooding) ou des attaquants (machines scannant le réseau par exemple). Enfin l'intérêt de mettre en place une détection de changements sur une structure de type *sketch* plutôt que sur le trafic agrégé est clairement mise en évidence, les courbes ROC étant excellentes dans le cas de la détection de changements sur *sketch* alors qu'une détection de changement sur le trafic agrégé conduit à de nombreuses fausses alarmes sans détecter les anomalies de faible intensité.

DI PIETRO Andrea, HUICI Felipe, BONELLI Nicola, TRAMMELL Brian, KASTOVSKY Petr, GROLEAT Tristan, VATON Sandrine, NUCCILLI Fabrizio, Blockmon : A High-Performance Composable Network Traffic Measurement System (soumis)

L'objectif de cet article est de présenter BlockMon [2], un système modulaire, flexible et à haute performance pour la conception d'applications de surveillance de trafic. Le travail de conception et de développement de Blockmon a été réalisé dans le cadre du projet européen *DEMONS : DEcentralized MONitoring for trustworthinesS* [9]. Les principaux concepteurs et développeurs de Blockmon sont l'université de Pise et ETH Zürich. La société tchèque INVEA-TECH ainsi que l'Institut Télécom se sont associés au travail de développement et de test de Blockmon dans le cadre du projet DEMONS. Blockmon est toujours en cours d'amélioration pour optimiser au maximum les performances du système mais il est depuis quelques mois disponible en OpenSource.

Le projet européen DEMONS qui a financé en partie ces travaux a pour objectif de permettre la mise en place de mécanismes coopératifs de détection et de réponse aux attaques à large échelle sur l'Internet. Pour cela une infrastructure de mesure décentralisée sur plusieurs domaines sous des juridictions différentes sera mise en place. Cela pose des problèmes de confidentialité des données de mesure et de confiance entre les différents domaines ainsi que des problèmes d'automatisation et de passage à l'échelle de la solution déployée au niveau intra et inter-domaine.

Le projet DEMONS est organisé en plusieurs *WorkPackages* (WP). Trois des WP correspondent à la décomposition du problème de détection et défense collaborative en couches : la couche de mesure (WP3), la couche coordination (WP4), et la couche applicative (WP5). La couche mesures (WP3) essaie de profiter des avancées récentes dans le domaine logiciel et matériel pour concevoir des points de mesure flexibles, programmables et passant à l'échelle. La couche coordination (WP4) est chargée de l'orchestration des mesures et des mécanismes de défense entre les différents domaines en respectant les contraintes de confidentialité des données. La couche application (WP5) utilise l'infrastructure mise en place par les deux autres couches pour développer et déployer rapidement des applications de mesure et de défense contre des menaces correspondant à différents scénarios (botnets, DDoS, attaques contre des serveurs

DNS, fraude en VoIP, etc...) Blockmon est développé dans le cadre du WP3 sur la couche mesure.

L'objectif de Blockmon est de permettre par sa structure modulaire et par l'utilisation de techniques d'optimisation en matériel et en logiciel le développement de points de capture et d'analyse de trafic flexibles et à haute performance. Blockmon combine la flexibilité pour permettre une large gamme d'analyses différentes du trafic avec les mécanismes nécessaires pour permettre la haute performance. Blockmon reprend certains principes de travaux sur le traitement modulaire et programmable de trafic comme le système de mesure CoMo [10] ou le routeur modulaire Click [14] mais propose un environnement flexible permettant une large gamme de traitements et optimise les traitements pour permettre une exécution haute performance sur des architectures multi-coeur.

A haut niveau, Blockmon fournit un ensemble de blocs génériques qui peuvent être composés pour réaliser une application particulière. La composition des blocs est spécifiée en XML et des scripts Python permettent de contrôler le système en temps réel. Le coeur de Blockmon et les blocs sont codés en C++. L'interconnexion entre les blocs se fait au travers de portes qui représentent des files de messages. Une composition particulière peut être déployée sur plusieurs coeurs de calcul ou même sur plusieurs machines différentes en utilisant la sérialisation/désérialisation des messages. Grâce à l'utilisation de buffers circulaires pour stocker les messages la composition peut être changée dynamiquement ou les blocs reconfigurés ce qui permet de s'adapter dynamiquement à des événements particuliers (changement de routage, raffinement progressif de l'exploration du trafic, etc...)

Les performances sont optimisées en utilisant au mieux le potentiel des technologies récentes sur le multi-coeur et les cartes réseau. Différentes stratégies sont exploitées : définition de pools de threads distribués sur les différentes unités de calcul, files circulaires sans blocage, allocations mémoires groupées, pointeurs partagés pour le passage des messages entre blocs, etc... Par ailleurs la capture des paquets est optimisée en permettant l'utilisation de cartes matérielles (COMBOv2 [12], NetFPGA [16]) ou une capture logicielle rapide exploitant PFQ [3] ou PF-RING [5] dont les performances dépassent largement la solution standard basée sur PCAP qui est également supportée. L'optimisation des performances du système a essentiellement été prise en charge par l'Université de Pise dans le cadre de la thèse d'Andrea di Pietro pour les aspects logiciels et par INVEA-TECH pour les aspects matériels. Différentes librairies ont été développées par Télécom Bretagne pour faciliter le développement des blocs : fonctions de hachage (MD5, SHA, etc...), structures de données probabilistes (Count Min Sketch, filtres de Bloom, etc...), détection de changements brusques (CUSUM paramétrique/non paramétrique).

Dans l'article quelques scénarios d'applications sont présentés. Le premier scénario très simple est une détection de flots massifs. Deux compositions sont présentées : l'une avec capture de paquets en matériel (carte COMBOv2), l'autre avec capture de paquets optimisée en logiciel avec PFQ. Sans optimisation particulière la composition en logiciel atteint des débits de 3,8 Gb/sec. tandis que celle en matérielle atteint le débit du lien de 10 Gb/sec. Le second scénario concerne la détection de TCP SYN flooding par la méthode d'identification de changements brusques sur des sketchs que nous avions développée lors de l'ANR OSCAR. Avec une approche logicielle les performances obtenues montrent la capacit d'analyser du trafic à un débit de 5.5 Gb/sec. avec 10% de paquets SYN. Le troisième scénario concerne la détection de fraudes en VoIP (telemarketing) et compare les performances d'une implémentation utilisant Blockmon avec l'implémentation initiale du détecteur. Une amélioration de 44% du débit de flots traités par seconde est obtenue en utilisant l'environnement Blockmon avec 4 coeurs de calcul : on passe de 46900 à 67600 appels traités par seconde.

Pour conclure, l'objectif de Blockmon est de combiner le meilleur de deux mondes : la flexibilité d'une architecture modulaire, et les performances de solutions matérielles et logicielles optimisées. Blockmon peut être étendu à plusieurs noeuds, supporter des applications concurrentes et reconfigurables dynamiquement pour une adaptation rapide des applications de mesure au trafic observé. Les travaux en cours concernent l'optimisation de l'allocation des blocs aux *pools* de *threads* en prenant en compte des indicateurs comme le CPU ou la mémoire utilisée pour améliorer au maximum les performances du système.

Références

[1] ANR RNRT (projet labellisé fin 2005). OSCAR : Overlay Security, Characterization, Analysis and Recovery.

[2] Blockmon consortium. Blockmon : a modular system for flexible, high-performance traffic monitoring and analysis. `http://blockmon.github.com/blockmon/`.

[3] N. Bonelli, A. Di Pietro, S. Giordano, and G. Procissi. Pfq : a novel engine for multi-gigabit packet capturing with multi-core commodity hardware. In *Proceedings of the Passive and Active Measurement Conference*, PAM 2012, 2012.

[4] Graham Cormode and S. Muthukrishnan. An Improved Data Stream Summary : the Count-Min Sketch and its Applications. *Journal of Algorithms*, 55(1) :58–75, April 2005.

[5] L. Deri. PFRING : high-speed packet capture, filtering and analysis. `http://www.ntop.org/products/pf_ring/`.

[6] Dr Peering (Bill Norton). DrPeering.net : The tools and talent needed to make strategic peering decisions. . `http://drpeering.net/`.

[7] Endace. Endace DAG high-speed capture cards. `http://www.endace.com/endace-dag-high-speed-packet-capture-cards.html`.

[8] J.M. Estevez-Tapiador, P. Garcia-Teodoro, and J.E. Diaz-Verdejo. Stochastic protocol modeling for anomaly based network intrusion detection. In *Proc. of IEEE International Workshop on Information Assurance*, IWIA'03, 2003.

[9] European Commission 7th Framework Program (FP7). DEMONS : DEcentralized MONitoring for trustworthinesS . `http://fp7-demons.eu/`.

[10] Intel Research Cambridge. The CoMo Project. http ://como.sourceforge.net/.

[11] Internet 2 Community (Abilene network). The Internet 2 observatory. `http://internet2.edu/observatory/`.

[12] INVEA-TECH. FPGA Solutions Overview. `http://www.invea-tech.com/fpga-solutions/fpga-solutions-overview`.

[13] W.H. Ju and Y. Vardi. A hybrid high-order markov chain model for computer intrusion detection, 1999.

[14] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The click modular router. *ACM Trans. Comput. Syst.*, 18 :263–297, August 2000.

[15] MADAM ID Project. DARPA Datasets. `http://www.ll.mit.edu/IST/ideval/data/data_index.html`.

[16] NetFPGA. NetFPGA : a line-rate, flexible, and open platform for research, and classroom experimentation. `http://netfpga.org/`.

[17] Robin Sommer and Vern Paxson. Outside the closed world : On using machine learning for network intrusion detection. *Security and Privacy, IEEE Symposium on*, 0 :305–316, 2010.

[18] Alexander Tartakovsky, Boris Rozovskii, Rudolf Blazek, and Hongjoong Kim. Detection of Intrusion in Information Systems by Sequential Chang-Point Methods. *Statistical Methodology*, 3(3) :252–340, 2006.

[19] University of New Mexico. System Calls Dataset. `http://www.cs.unm.edu/~immsec/systemcalls.htm`.

[20] Université de Pise, Italie. Département d'Ingénierie de l'Information. `http://dip-iet.iet.unipi.it/`.

[21] Université de Pise, Italie. Telecommunication Networks Research Group. `http://wwwtlc.iet.unipi.it/`.

[22] Y. Vardi and D. Lee. From image deblurring to optimal investments : Maximum likelihood solutions for positive linear inverse problems. *Journal of the Royal Statistical Society B*, 55 :569–612, 1993.

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Eur. Trans. Telecomms. 2010; **21**:575–588
Published online 15 June 2010 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/ett.1432

Communication Networks

A new statistical method for detecting network anomalies in TCP traffic

Christian Callegari[1]*, Sandrine Vaton[2] and Michele Pagano[1]

[1]*Department of Information Engineering, University of Pisa, Pisa, Italy*
[2]*Department of Computer Science, TELECOM Bretagne, Brest, France*

SUMMARY

In the last few years, the number and impact of security attacks over the Internet have been continuously increasing. To face this issue, the use of Intrusion Detection Systems (IDSs) has emerged as a key element in network security. In this paper we address the problem considering a novel statistical technique for detecting network anomalies. Our approach is based on the use of different families of Markovian models, namely high order and non-homogeneous Markov chains, for modeling network traffic running over TCP. The performance results shown in the paper justify the proposed method and highlight the improvements over commonly used statistical techniques. Copyright © 2010 John Wiley & Sons, Ltd.

1. INTRODUCTION

In the last few years Internet has experienced an explosive growth. Along with the wide proliferation of new services, the quantity and impact of attacks have been continuously increasing. The number of computer systems and their vulnerabilities have been rising, while the level of sophistication and knowledge required to carry out an attack have been decreasing, as much technical attack know-how is readily available on Web sites all over the world (see the CERT research annual reports [1]).

Recent advances in encryption, public key exchange, digital signature and the development of related standards have set a foundation for network security. However, security on a network goes beyond these issues. Indeed it must include security of computer systems and networks, at all levels, top to bottom.

Since it seems impossible to guarantee complete protection to a system by means of prevention mechanisms (e.g. authentication techniques), the use of an Intrusion Detection System (IDS) is of primary importance to reveal intrusions in a network or in a system. IDSs are usually classified on the basis of several criteria [2, 3]. A first classification is based on the system scope, distinguishing between host based IDS (HIDS) and network based IDS (NIDS). A second classification divides the systems in stateless IDSs (analyse each event independently of the others) and stateful IDSs (maintain information about the past events: the effect of a certain event depends on its position in the event stream). At last, the most important classification criterion is based on the detection technique used to reveal the intrusions. In this case IDSs are divided in:

- misuse based (or signature based) IDSs: identify intrusions by watching for patterns of traffic or application data presumed to be malicious. This type of systems is only able to detect 'known' attacks, described by a set of signatures.
- anomaly based IDSs: detect intrusions by monitoring system activity and classifying it as either normal or anomalous. The classification is based on heuristics or rules, rather than patterns or signatures. The IDS detects any type of *misuse* that falls out of normal system behaviour. An anomaly based IDS should be able to detect new attacks, but produces much more false alarms than a misuse based IDS.

* Correspondence to: Christian Callegari, Department of Information Engineering, University of Pisa, Pisa, Italy. E-mail: christian.callegari@iet.unipi.it

Received 31 July 2009
Revised 8 March 2010
Accepted 11 April 2010

State of the art in the field of intrusion detection is mostly represented by misuse based IDSs. Considering that most attacks are realised with known tools, available on the Internet, a signature based IDS could seem a good solution. Nevertheless hackers continuously come up with new ideas for the attacks, that a misuse based IDS is not able to block. This is the main reason why our work has focused on the development of an anomaly based IDS. In particular our goal is to reveal intrusions carried out exploiting TCP bugs, by using Markovian models (high order and non-homogeneous Markov chains) to describe the behaviour of the network traffic.

The paper is structured as follows. The next section presents the related works. The section thereafter provides a detailed description of the implemented system; while the subsequent section presents the experimental results. Finally, the paper concludes with some final remarks.

2. RELATED WORKS

The use of a statistical approach to detect anomalies in the network traffic was first introduced by Denning in Reference [4]. The author proposed an early, abstract model of an Intrusion Detection Expert System (IDES), based on the statistical characterisation of the behaviour of a *subject* with respect to a given *object*. The basic idea was to realise a *profile* of the normal behaviour of the system. Denning proposed several statistical models to be used to build the *profile*; among these she considered a *Markov Process Model* to describe the transition probabilities for a given *metric*. Since then, lots of papers have taken into account the use of first order homogeneous Markov chains to detect two distinct kinds of 'anomalies': masqueraders (analysing the command stream of a host) and intruders (analysing the evolution of TCP flows in the network traffic) [5].

Vardi and Ju in Reference [6] described the use of high order Markov chains to detect masqueraders at the host level, and in References [7, 8] the authors compared performance of first order models and 'generic' high order models.

After an extensive survey, to the best of our knowledge, there is no work directly related neither to the use of high order Markov chains to detect anomalies in the TCP traffic nor to the application of non-homogeneous Markov chains to anomaly detection in general.

Moreover no study at all compares the performance achievable with Markov chains of different orders and with a simple 'independent' model, which just takes into account state occurrence probabilities.

3. SYSTEM DESIGN

The aim of our work is to perform a comparison between several statistical models, which can be used to describe the behaviour of TCP connections. More in detail we take into account the use of:

- first order homogeneous Markov chains
- first order non-homogeneous Markov chains
- high order homogeneous Markov chains
- stationary ECDF (Empirical Cumulative Distribution Function)
- non-stationary ECDF

Next subsections describe the training phase and the detection phase of our IDS.

3.1. Training phase

To build the model which represents the 'normal' behaviour of the network, the system needs a training phase during which it analyses some network traffic, supposed to be attack free. The system analyses raw traffic traces in libpcap format [9], the standard used by publicly available packet sniffer software, as Tcpdump, Wireshark or Snort. First of all the IDS performs a filtering phase so that only TCP packets are passed as input to the detection blocks.

The IDS only considers some fields of the packet headers, more precisely the IP source address, the IP destination address, the source port number, the destination port number and the TCP flags. The IP addresses and the port numbers are used to identify a connection, while the value of the flags is used to build the profile. Experimental results have shown that the stochastic models associated to the traffic directed to different port numbers strongly differ one from the other. Thus, the following procedure is realised once per each port number. After that the IDS reconstructs the single connections on the basis of the five-tuple (source and destination addresses, source and destination ports and protocol).

A value s_i is associated to each packet, according to the configuration of the TCP flags:

$$s_i = syn + 2 \cdot ack + 4 \cdot psh + 8 \cdot rst + 16 \cdot urg + 32 \cdot fin \quad (1)$$

Thus each 'mono-directional' connection is represented by a sequence of symbols s_i, which are integers in $\{0, 1, \cdots, 63\}$.

DOI: 10.1002/ett

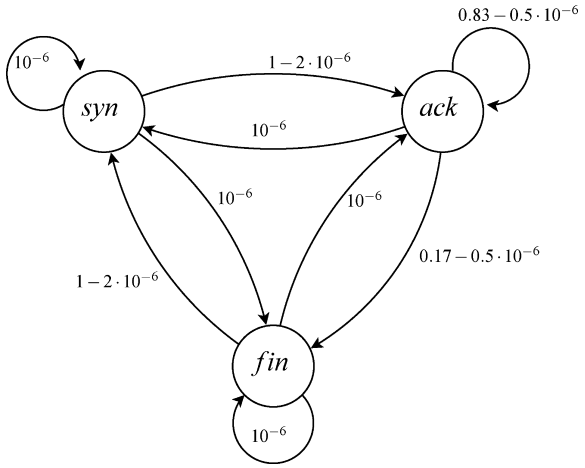


Figure 1. Example of a Markov chain associated to a given application.

The training phase, as well as the detection phase, varies according to the stochastic model we are taking into account.

3.1.1. ECDF. In the case of the stationary ECDF the training phase simply consists of evaluating the probabilities $P(s_i)$ that the TCP flags assume the value s_i, independently of the position of the packet in the TCP connection.

For the non-stationary ECDF the system has to compute the probabilities $P_j(s_i)$ that the TCP flags of the jth packet of the connection assume the value s_i. Taking into account the nature of the security attacks, for reducing the complexity of the system, we have decided to evaluate such probabilities only for the first 10 packets of each connection, i.e. $j = 1, 2, \cdots, 10$.

3.1.2. Markov chains. In the case of Markovian models the symbols s_i are considered as the states of a discrete time finite state Markov chain.

Before detailing the proposed algorithms, we would like to point out, with a naive example, the rationale behind the use of Markov chains for anomaly detection. For this purpose, let us consider the Markov chain depicted in Figure 1 and let us suppose to observe the sequence $S_1 =$ *syn*, *ack*, *ack*, *ack*, *ack*, *fin*, corresponding to the normal behaviour of the considered application, and the sequence $S_2 =$ *syn*, *syn*, *syn*, *syn*, *syn*, *syn*, which corresponds to a SYN flooding attack. Then, let us suppose that we decide if a connection is anomalous or not on the basis of the product of the transition probabilities associated to the given sequence. It is easy to realise that the first connection will take to a relatively high value, while the second will correspond to a very low value of the computed product. Thus it will be enough to set a threshold and decide that a connection is anomalous if the associated product of the transition probabilities does not overcome that threshold.

Since not all the TCP flags configurations are present in real traffic, the system only considers the states observed in the training phase. Moreover, to take into account the possibility that some new flags configurations could be observed during the detection phase, a rare state is added. This procedure allows us to reduce the cardinality of the state space from 64 (all the possible configurations of the six TCP flags bits) to a number K, usually smaller than 10.

Then the system estimates the transition probabilities of the Markov chain. A Markov chain is characterised by the well-known weak Markov property:

- first order non-homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) = P(C_t = s_{i_0} | C_{t-1} = s_{i_1}) = P_t(s_{i_0} | s_{i_1}) \quad (2)$$

- lth order homogeneous Markov chain

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, C_{t-3} = s_{i_3}, \cdots) = P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, \cdots, C_{t-l} = s_{i_l}) = P(C_0 = s_{i_0} | C_{-1} = s_{i_1}, C_{-2} = s_{i_2}, \cdots, C_{-l} = s_{i_l}) = P(s_{i_0} | s_{i_1}, s_{i_2}, \cdots, s_{i_l}) \quad (3)$$

where C_t represents the state of the chain at step t.

Since the computation of such probabilities is quite straightforward in the case of first order Markov chains (homogeneous and non-homogeneous), in the following we consider a Markov chain of order l. The main problem related to this kind of models is the 'explosion' of the number of parameters, which grows exponentially with the order, according to the rule $K^l(K-1)$. This entails the need of a parsimonious representation of the transition probabilities. The approach used in this paper is the Mixture Transition Distribution (MTD) model, first proposed in Reference [10]. Under the MTD model, the transition probabilities of an lth order Markov chain can be expressed as follows:

$$P(C_t = s_{i_0} | C_{t-1} = s_{i_1}, C_{t-2} = s_{i_2}, \cdots, C_{t-l} = s_{i_l}) = \sum_{j=1}^{l} \lambda_j r(s_{i_0} | s_{i_j}) \quad (4)$$

where the quantities

$$\begin{aligned}\mathbf{R} &= \{r(s_i|s_j);\ i, j = 1, 2, \cdots, K\}\\ \mathbf{\Lambda} &= \{\lambda_j;\ j = 1, 2, \cdots, l\}\end{aligned} \tag{5}$$

satisfy to the following constraints:

$$\begin{aligned} r(s_i|s_j) &\geqslant 0 \qquad i, j = 1, 2, \cdots, K\\ \textstyle\sum_{s_i=1}^{K} r(s_i|s_j) &= 1 \qquad \forall j = 1, 2, \cdots, K\end{aligned} \tag{6}$$

$$\begin{aligned} \lambda_j &\geqslant 0 \qquad j = 1, 2, \cdots, l\\ \textstyle\sum_{j=1}^{l} \lambda_j &= 1\end{aligned} \tag{7}$$

A consequence of the use of the MTD model is the reduction of the number of parameters from $K^l(K-1)$ to $K(K-1)+l-1$. According to the MTD model, the log-likelihood of a sequence $(c_1, c_2, \cdots, c_T)$ of length T is

$$LL(c_1, c_2, \cdots, c_T) = \sum_{i_0=1}^{K} \cdots \sum_{i_l=1}^{K} N(s_{i_0}, s_{i_1}, \cdots, s_{i_l}) \cdot \log\left(\sum_{j=1}^{l} \lambda_j r(s_{i_0}|s_{i_j})\right) \tag{8}$$

where $N(s_{i_0}, s_{i_1}, \cdots, s_{i_l})$ represents the number of times the transition $s_{i_l} \to s_{i_{l-1}} \to \cdots \to s_{i_0}$ is observed. Maximum likelihood estimation (MLE) of the chain parameters requires to maximise the right hand side of Equation (8), with respect to $\mathbf{R}$ and $\mathbf{\Lambda}$, taking into account the constraints (6) and (7).

Since the original solution [11] seems to be too much computationally demanding, we have applied the procedure proposed in Reference [6], which consists in an alternate maximisation with respect to $\mathbf{R}$ and to $\mathbf{\Lambda}$. This process leads to a global maximum, since LL is concave in $\mathbf{R}$ and $\mathbf{\Lambda}$. For the part when $\mathbf{R}$ is fixed, we maximise LL with respect to $\mathbf{\Lambda}$, and *vice versa*. In the first step (estimation of $\mathbf{\Lambda}$) we have used the sequential quadratic programming, while the second maximisation step (estimation of $\mathbf{R}$) is a linear inverse problem with positivity constraints (LININPOS) that we have solved applying the expectation maximisation (EM) algorithm [12]. Since the first maximisation step is quite trivial, in what follows we discuss the second step, i.e. the estimation of the matrix $\mathbf{R}$, with the vector $\mathbf{\Lambda}$ fixed. First of all we have re-indexed the log-likelihood in the following way:

$$\phi(s_{i_0}, s_{i_1}, \cdots, s_{i_l}) \stackrel{\Delta}{=} 1 + \sum_{j=0}^{l} (s_{i_j} - 1)K^{l-j} \to k \tag{9}$$

which takes to

$$N(s_{i_0}, s_{i_1}, \cdots, s_{i_l}) \to a_k \tag{10}$$

$$\sum_{j=1}^{l} \lambda_j r(s_{i_0}|s_{i_j}) \to b_k \tag{11}$$

Thus, at first we estimate the quantities b_k and then we solve the linear system

$$b_k = \sum_{j=1}^{l} \lambda_j r(s_{i_0}|s_{i_j}) \tag{12}$$

which is a LININPOS problem.

At this point, the log-likelihood can be expressed as:

$$\sum_{k=1}^{K^{l+1}} a_k \log b_k \tag{13}$$

where Reference [6]

$$\sum_{k=1}^{K^{l+1}} a_k = T - l \tag{14}$$

and

$$\sum_{k=1}^{K^{l+1}} b_k = K^l \tag{15}$$

Thus a simple Lagrange method argument shows that the log-likelihood is maximised when

$$\hat{b}_k = \frac{a_k}{\sum_k a_k} \sum_k b_k = \frac{a_k}{T-l} K^l \qquad \forall k \tag{16}$$

or, equivalently, when

$$\sum_{j=1}^{l} \lambda_j r(s_{i_0}|s_{i_j}) = \frac{K^l}{T-l} N(s_{i_0}, s_{i_1}, \cdots, s_{i_l}) \quad \forall (i_0, \cdots, i_l) \tag{17}$$

Thus, if we consider these equations as a linear system subject to the constraints (6), we obtain a LININPOS problem, which can be solved, in the sense of the minimum Kullback–Leibler distance, using the EM algorithm. More in detail we have Reference [13]

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \tilde{\mathbf{R}} = \begin{pmatrix} \frac{K^l}{T-l} \cdot \mathbf{N} \\ \mathbf{1} \end{pmatrix} \tag{18}$$

where

$$\begin{aligned} \tilde{\mathbf{R}}^T &= (r(s_1|s_1), r(s_2|s_1), \cdots, r(s_K|s_1), \cdots, r(s_K|s_K)) \\ &= (r_1, r_2, \cdots, r_{K^2}) \end{aligned} \tag{19}$$

are the unknowns, $r(s_i|s_j) = r_{i+K(j-1)}$, and

$$\mathbf{N} = \begin{pmatrix} N(s_1, s_1, \cdots, s_1) \\ N(s_1, s_1, \cdots, s_2) \\ \vdots \\ N(s_1, s_1, \cdots, s_K) \\ \vdots \\ N(s_K, s_K, \cdots, s_1) \\ \vdots \\ N(s_K, s_K, \cdots, s_K) \end{pmatrix} = \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_K \\ \vdots \\ N_{1-K+K^{l+1}} \\ \vdots \\ N_{K^{l+1}} \end{pmatrix} \tag{20}$$

where $N(s_0, s_1, \cdots, s_l) = N_i,\ i = \phi(s_{i_0}, s_{i_2}, \cdots, s_{i_l})$

$$\begin{gathered} \mathbf{A} = \{a_{ij}\}_{K^{l+1} \times K^2} \\ \text{where } a_{ij} = \textstyle\sum_{k=0}^{l} \lambda_k I[j = i_0 + K(i_k - 1)], \\ (i_0, \cdots, i_l) = \phi^{-1}(i) \end{gathered} \tag{21}$$

Thus the matrix **A** and **B** look like Equation (22) and (23), while **1** is a vector of ones:

$$\mathbf{A} = \begin{pmatrix} \lambda_1 + \cdots + \lambda_l & 0, \cdots 0 & 0 & 0, \cdots, 0 & 0 & 0, \cdots, 0 \cdots & 0 \\ \lambda_1 + \cdots + \lambda_{l-1} & 0, \cdots 0 & \lambda_l & 0, \cdots, 0 & 0 & 0, \cdots, 0 \cdots & 0 \\ \lambda_1 + \cdots + \lambda_{l-1} & 0, \cdots 0 & 0 & 0, \cdots, 0 & \lambda_l & 0, \cdots, 0 \cdots & 0 \\ & & & \vdots & & & \\ 0 & 0, \cdots 0 & 0 & 0, \cdots, 0 & 0 & 0, \cdots, 0 \cdots & \lambda_1 + \cdots + \lambda_l \end{pmatrix}_{K^{l+1} \times K^2} \tag{22}$$

$$\mathbf{B} = \begin{pmatrix} 1, \cdots, 1 & 0, \cdots 0 & 0, \cdots 0 & 0, \cdots, 0 \\ 0, \cdots 0 & 1, \cdots, 1 & 0, \cdots 0 & 0, \cdots, 0 \\ 0, \cdots 0 & 0, \cdots 0 & \ddots & 0, \cdots, 0 \\ 0, \cdots 0 & 0, \cdots 0 & 0, \cdots 0 & 1, \cdots, 1 \end{pmatrix}_{K \times K^2} \tag{23}$$

$$\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{K \times 1} \tag{24}$$

At this point the EM iteration step is the following:

$$\begin{aligned} r_j \leftarrow \frac{a_{\cdot j}}{a_{\cdot j} + b_{\cdot j}} \hat{r}_j \left(\mathbf{A}, \frac{K^l}{T-l} \cdot \mathbf{N}, \tilde{\mathbf{R}} \right) \\ + \frac{b_{\cdot j}}{a_{\cdot j} + b_{\cdot j}} \hat{r}_j (\mathbf{B}, \mathbf{1}, \tilde{\mathbf{R}}) \end{aligned} \tag{25}$$

where

$$\begin{gathered} \hat{r}_j(\mathbf{W}, \mathbf{u}, \mathbf{v}) \equiv \frac{v_j}{w_{\cdot j}} \sum_i \frac{w_{ij} u_i}{\sum_k w_{ik} v_k} \\ j = 1, 2, \cdots, K^2 \end{gathered} \tag{26}$$

for matrix $\mathbf{W} = \{w_{ij}\}$ and vectors $\mathbf{u} = \{u_i\}$, $\mathbf{v} = \{v_i\}$, and

$$\begin{aligned} a_{\cdot j} &= \sum_i a_{ij} = \sum_{(i_0, \cdots, i_l) = \phi^{-1}(i)} \sum_{k=0}^{l} \lambda_k I[j = i_0 + K(i_k - 1)] \\ &= \sum_{k=0}^{l} \lambda_k \sum_{i_0} \cdots \sum_{i_l} I[j = i_0 + K(i_k - 1)] \\ &= \sum_{k=0}^{l} \lambda_k K^{l-1} = K^{l-1} \end{aligned} \tag{27}$$

and $b_{\cdot j} = \sum_i b_{ij} = 1$.

DOI: 10.1002/ett

The choice of the initial values for $\mathbf{R}$ and $\mathbf{\Lambda}$ is a key point. Experimental tests have shown that good results are obtained choosing $\lambda_i = 1/l,\ i = 1, 2, \cdots, l$. As far as $\mathbf{R}$ is concerned the quantities $r(s_i|s_j)$ are set equal to the first order transition probabilities, plus the following quantities, which take into account the presence of the 'rare' state (labelled K):

$$\begin{gathered} r(rare|s_i) = \epsilon \qquad \forall i = 1, 2, \cdots, K \\ \text{with } \epsilon \text{ small} \\ r(s_i|rare) = (1-\epsilon)/(K-1) \\ \forall i = 1, 2, \cdots, K-1 \end{gathered} \tag{28}$$

In our case $\epsilon = 10^{-6}$; to be noted that, if the training is correctly performed, it should be sufficient to choose this probability, such that it is much lower then the others.

3.2. *Detection phase*

Once the training phase has been performed, the IDS has a model of the 'normal' behaviour of the network, represented by the computed *profile*.

As for the training phase, the input is given by raw traffic traces in libpcap format, which are processed so as to extract sequences of TCP flags configurations.

Thus, given an observed sequence $(c_1, c_2, \cdots, c_T)$, the system has to decide between the two hypotheses:

$$\begin{gathered} H_0 : \{(c_1, c_2, \cdots, c_T) \sim \text{computed model}\} \\ H_1 : \{\text{anomaly}\} \end{gathered} \tag{29}$$

The problem is to choose between a single hypothesis H_0, which is associated to the estimated stochastic model (Markov chain of parameter $\theta_0 = \{\mathbf{\Lambda}_u, \mathbf{R}_u\}$) and the composite hypothesis H_1, which represents all the other possibilities (described by a Markov chain of parameter $\theta \in \Theta, \theta \neq \theta_0$). No optimal result is presented in the literature about this decision making problem, thus the best solution is represented by the use of the Generalised Likelihood Ratio (GLR) test [14].

Since the problem is quite straightforward for ECDF, in the following we only consider the case of Markovian models, for which the GLR test is defined as follows:

$$H(X) = \begin{cases} H_0 \ if \ X > \xi \\ H_1 \ if \ X < \xi \end{cases} \tag{30}$$

where ξ is an appropriate threshold and the quantity X is given by:

$$X = \left(\frac{L(c_1, c_2, \cdots, c_T|\theta_0)}{Max_{\theta \neq \theta_0} L(c_1, c_2, \cdots, c_T|\theta)} \right)^{\frac{1}{T}} \tag{31}$$

where L represents the likelihood function of the sequence and the component $\frac{1}{T}$ takes into account the length T of the observed sequence.

It is worth noticing that this test is equivalent to decide on the basis of the Kullback–Leibler divergence between the model associated to H_0 and the one computed for the observed sequence. The Kullback–Leibler divergence, for first order Markov chains, is defined as:

$$KL\,(\mathrm{MC}_s, \mathrm{MC}_0) = \sum_i \sum_j \pi_s(s_i) P_s(s_j|s_i)\ \log \frac{P_s(s_j|s_i)}{P_0(s_j|s_i)} \tag{32}$$

where $\pi_0(s_i)$ is the stationary distribution of MC_0 (i.e. $\pi_0(\cdot)$ such that $\pi_0 \cdot P_0 = \pi_0$) and $P_k(s_j|s_i)$ is the (single step) transition probability from state $C_{t-1} = s_i$ to state $C_t = s_j$ for the Markov chain MC_k (where $k = 0, 1, \ldots, S$).

Indeed, if MC_s represents the Markov chain model with parameter $\hat{\Theta}$ (ML estimation), then $-log(X)$ corresponds to the KL divergence between MC_s and MC_0.

The previous formula for the evaluation of the Kullback–Leibler divergence can be easily extended to Markovian models of order l. In this case the future evolution of the system at time t is defined by Equation (3).

This means that the basic definition for first-order Markov chains (with K states) may be extended considering the state of the chain $\boldsymbol{C_t}$ as a point in a finite l-dimensional lattice:

$$\boldsymbol{C_t} = (C_t, C_{t-1}, \ldots, C_{t-l+1})$$

Note that in this case the transition probability matrix is sparse, since at most K out of K^l transitions are allowed out of each states (i.e. the first $l-1$ components of $\boldsymbol{C_{t-1}}$ must be equal to the last $l-1$ components of $\boldsymbol{C_t}$).

4. EXPERIMENTAL RESULTS

In this section we compare the performance of the different statistical models over two distinct data sets. The first performance analysis has been carried out over the DARPA data set, while for the second one we have used a data set

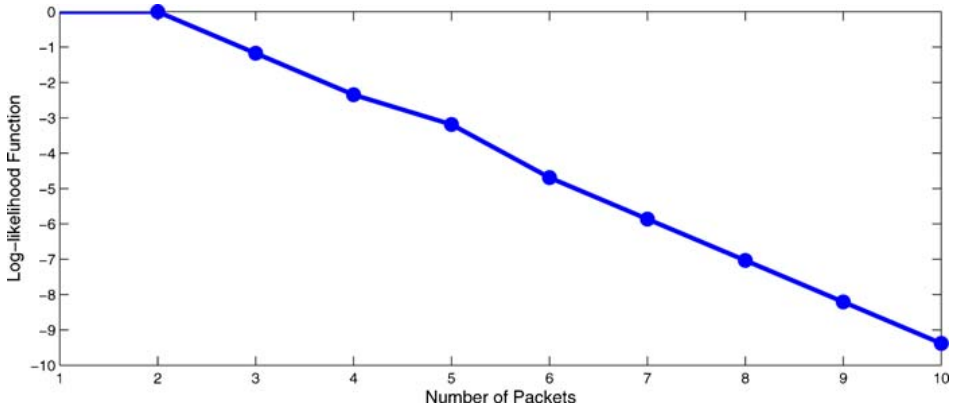


Figure 2. Log-likelihood function of a 'normal' connection.

expressly collected at the laboratory of the telecommunication network research group of the university of Pisa.

It is worth noticing that we have chosen to test the system over two distinct data sets, because as explained in the following subsections, they present different characteristics.

To test the correctness of the computed models we have calculated the log-likelihood function of some sequences. Figure 2 corresponds to a 'normal' connection (first 10 packets of a connection, described by the following flags sequence: syn, ack, ack, ack, ack, ack, ack, ack, ack ,ack). As expected from the theory, the function decreases almost linearly with the number of packets; its slope is equal to the entropy of the model, which, for first order Markov chain, is defined as:

$$H(\mathrm{MC}) = \sum_i \sum_j \pi(i) P(s_j|s_i) \log P(s_j|s_i) \qquad (33)$$

where $\pi(i)$ is the stationary distribution of the Markov chain. The given definition can be easily extended for higher order Markov chains, following the procedure described in the previous section for the Kullback–Leibler divergence.

On the other hand the effect of an anomaly is an abrupt jump in the log-likelihood function, as highlighted by Figure 3. It is worth noticing that in this case we have considered a TCP connection described by the flags sequence syn, ack, ack, ack, ack, ack, rst, ack, ack, ack, which corresponds to a simple scenario, where a hacker performs a Man-in-the-Middle attack, to withdraw an established connection.

To be noted that, both these figures refer to a first order model, but the behaviour of the log-likelihood function does not significantly vary with the order of the Markov chain.

To evaluate the performance we have used the Receiver Operating Characteristic (ROC) curves, which plot detection rate *versus* false positive rate, obtained varying the value of the threshold ξ.

4.1. DARPA data set

The DARPA evaluation project [15–17] represents the first formal, repeatable and statistically-significant evaluation of

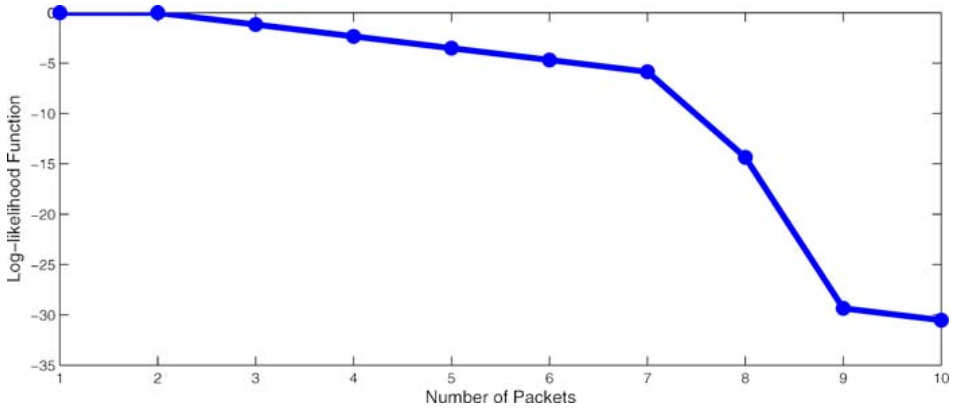


Figure 3. Log-likelihood function of an 'anomalous' connection.

DOI: 10.1002/ett

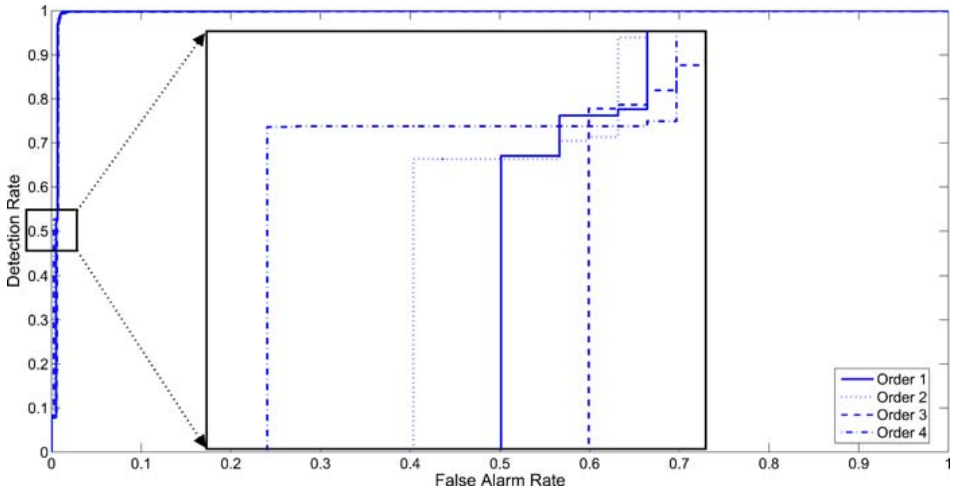


Figure 4. Performance of homogeneous Markovian models of different orders (data set 1).

IDSs and is considered the reference benchmark for testing this kind of systems.

Such project was carried out in 1998 and 1999, and the results shown in this paper have been obtained with the 1999 data set. This data set consists of five weeks of traffic, collected over a network composed by about 40 computers. The first and the third weeks do not contain any attack, thus they are suitable to perform the training of anomaly detection systems. The second week of traffic contains some labelled attacks and can be used to analyse how they modify network behaviour. Finally traffic from the fourth and the fifth weeks are the *test data*. There are 201 instances of 56 types of attacks distributed throughout these two weeks. Information about the attack instances (e.g. where they are located in week 4 and 5 data) is found in the 1999 'Attack Truth list', which is provided along with the data set. Moreover two distinct data sets are provided, respectively collected on the external and the internal side of the LAN gateway. In this work we have considered the latter, where the total number of attack instances is 177.

This data set could be considered too old and inadequate for being used to test new proposed IDSs. Nevertheless, the DARPA data set is still used, because it gives the possibility to produce results that can be easily replicated and verified by everyone.

For sake of brevity, in the following we only present the results related to the Telnet traffic, since they appear to be representative of the overall performance. Figure 4 shows the ROC curves for Markov chains of different orders. Since the results obtained using a model based on a Markov chain of order 1 are already very good for these traffic traces, it is not easy to realise that we achieve some improvements with higher order models. To be noted that the ROC curves are almost ideal, since we have a detection rate close to 100% with a negligible false alarm rate. Nevertheless the zoomed area inside the figure shows that with the model of order 4 we are able to achieve the best results, obtaining a detection rate of 53% with a false alarm rate which is about one half of that related to the Markov chain of order 1.

The following three figures show the performance of the other statistical models analysed in the paper. Figure 5 refers to non-homogeneous first order Markov chains, while Figures 6 and 7 respectively show the results achieved with stationary and non-stationary ECDF.

As it appears clear we cannot directly compare these performance with those related to the use of homogeneous Markov chains, since a detection performed analysing only the first 10 packets of each connection is obviously worse than the one based on the entire connections. For this reason in Figure 8 we present the performance comparison between time-dependent models and first order homogeneous Markov chains, analysing only the first 10 packets of each connection. It is easy to conclude that the homogeneous Markov chain achieves a detection rate almost 10% bigger than the other two models.

This apparent paradox can be justified by the fact that the non-homogeneous models have been computed with a relatively short, and so 'incomplete', training phase. Indeed, on one side the whole training data set has been used to compute only one homogeneous model, while on the other side, the same quantity of data is partitioned into 10 subsets corresponding to the first 10 steps in the time evolution of each connection. In particular this can lead to almost deterministic probabilities for the first steps of the non-homogeneous

DOI: 10.1002/ett

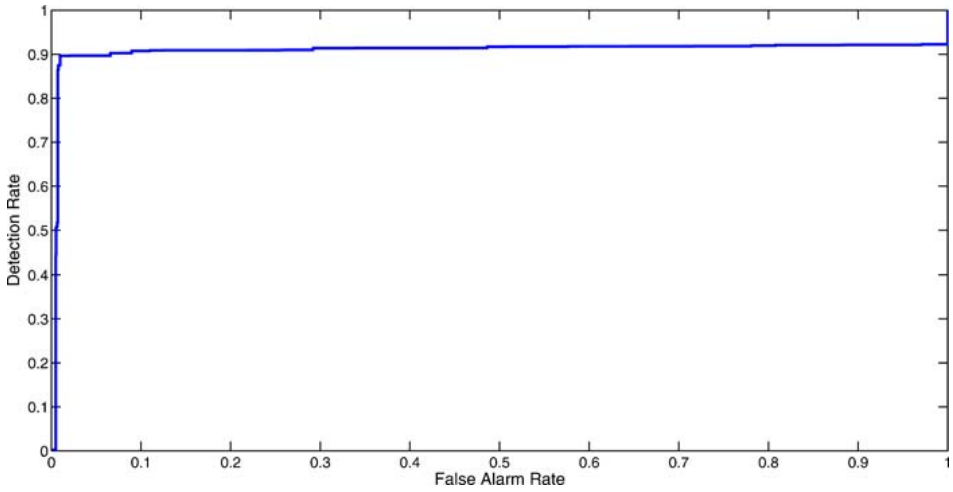


Figure 5. Performance of non-homogeneous Markov chain model (data set 1).

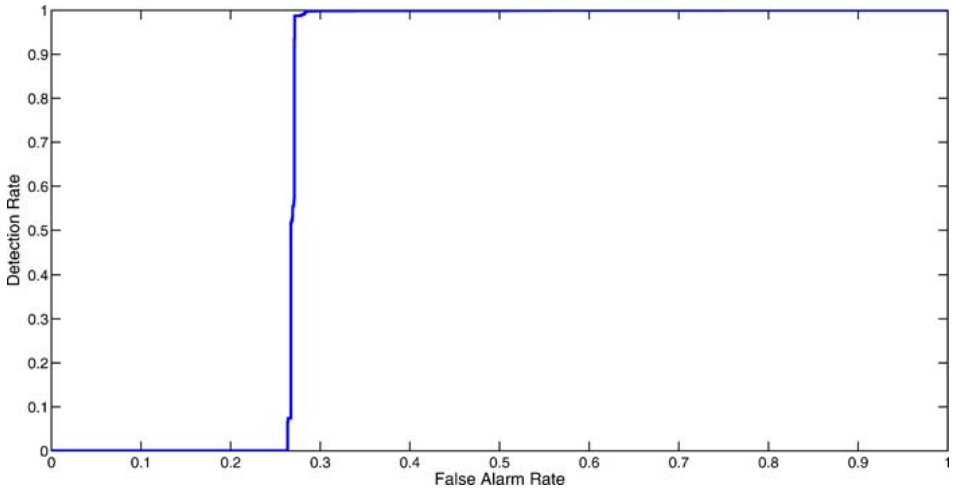


Figure 6. Performance of ECDF model (data set 1).

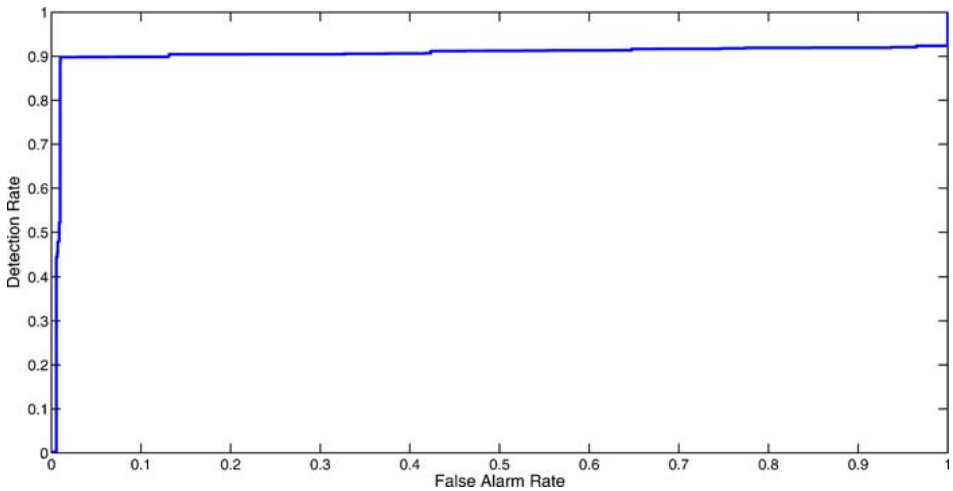


Figure 7. Performance of non-stationary ECDF model (data set 1).

DOI: 10.1002/ett

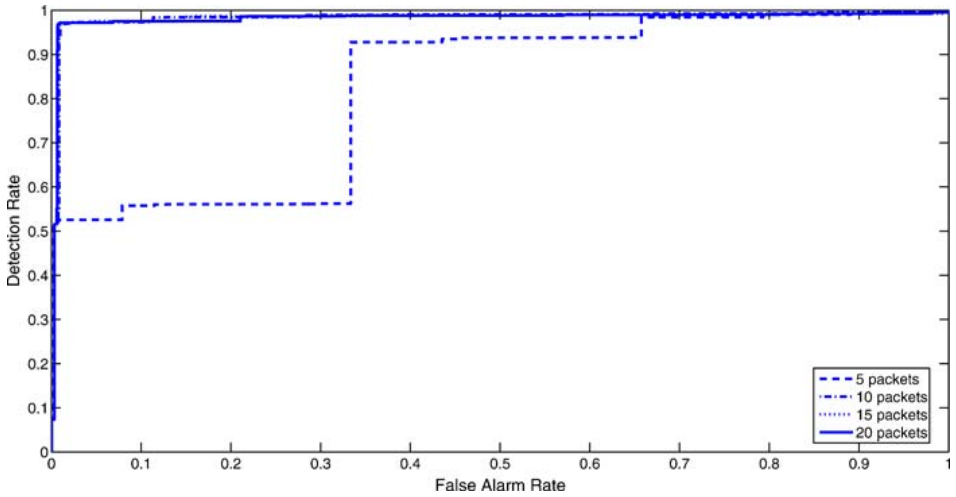


Figure 8. Performance comparison of the analysed time dependent models *versus* homogeneous Markov chain (10 packets only) (data set 1).

models, thus a single flag configuration at step i, present in the training data set only at steps $j \neq i$ (and hence captured by the time independent model), may generate a false alarm.

Finally, we have taken into account that an intrusion should be detected as soon as the anomaly appears. Thus, in Figure 9, we show the performance of first order homogeneous Markov chain model as a function of the number of analysed packets for each connection (both for building the model and for the detection phase). The results highlight that good performance are achieved with a small number of packets, demonstrating that such statistical models are suitable for on line anomaly detection.

4.2. University of Pisa data set

The second data set has been expressly collected at the LAN gateway of the laboratory of the telecommunication network research group of the university of Pisa, which is composed of about 30 PCs, running the most common operating systems (Windows, Linux and Mac OS).

In fact, for the training phase we still have used the training data set provided by the DARPA project, while the detection phase has been carried out using our own data set. This has been done so as to demonstrate the robustness of the proposed system. Moreover in this way, we have pointed out that everyone can correctly train the IDS using a public data set.

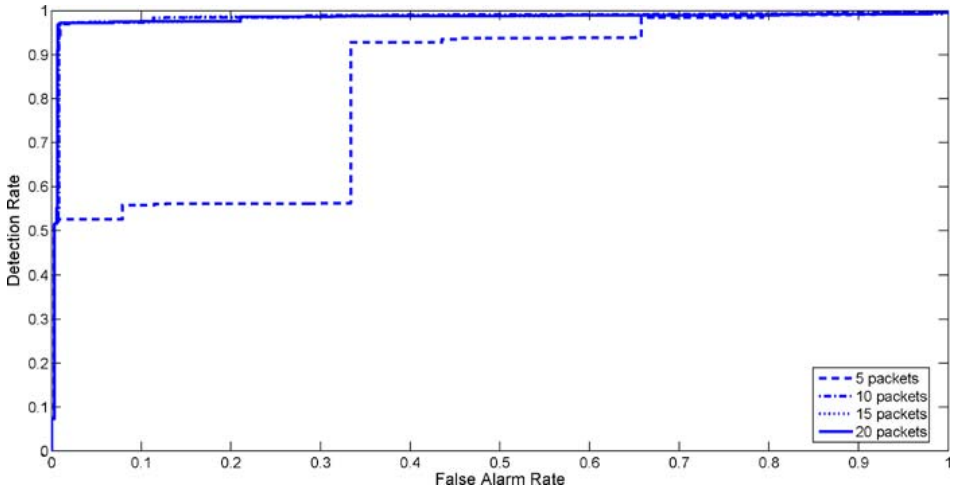


Figure 9. Performance of first order homogeneous Markov chain model, as a function of the number of processed packets (data set 1).

DOI: 10.1002/ett

Table 1. Attacks in data set 2.

Attack type	Attack name	No. of Attacks
ICMP flood	Smurf	
	Ping flood	28
	Ping of death	
Teardrop	-	13
TCP flood	SYN flood	
	Malformed packets	21
Password guessing	SSH password guessing	5
Probing	Port scan	
	Network scan	
	OS guessing	48
	Vulnerability scan	
	Service scan	
TOTAL		115

Table 2. Detection rate.

Attack type	Detection rate
ICMP flood	28/28
Teardrop	10/13
TCP flood	18/21
Password guessing	4/5
Probing	44/48

In more detail our data set is composed of traffic collected over a week, containing thousands of distinct connections. To allow the testing of the system, we have performed more than 100 distinct attacks, by using around 40 different open-source tools, which represent the state-of-the-art in the field of network security attacks (see Table 1 for the details). In this way we have been able to test the system with different types of attacks (including, among the others, DoS attacks, probe attacks and brute-force attacks). Moreover, by means of a manual inspection of the traffic traces we have been able to correctly label all the performed attacks, also verifying that no other attacks were present.

It is worth noticing that, different from the DARPA data set, our data includes traffic generated by multimedia applications (e.g. VoIP applications and Skype), as well as P2P applications (e.g. eMule/aMule and torrent). This implies that the server port number cannot be considered reliable anymore, for traffic classification purposes. In more detail, we have verified that traffic directed to port 80 represents a mixture of traffic, generated by several applications. Moreover, since we have collected real traffic (to be noted that, in fact, the DARPA traffic is synthetically generated), some losses and retransmission are present in the data.

For sake of brevity, in the following we only present the results related to the port 80 traffic. This choice is justified by the fact that, because of the presented nature of the traffic, this can be considered a sort of 'worst case' testing scenario.

Figure 10 shows the ROC curves for Markov chains of different orders. It is important to point out that the ROC curves present some 'steps', due to the fact that some attacks are characterised by a huge amount of similar connections, which are thus detected, in correspondence of the same value of the threshold.

As expected, the use of real traffic takes to a slightly worsening of the achieved results, nevertheless the system is still able to obtain a very good detection rate (about 90%) with a low false alarm rate (about 10%). Moreover, differently from the previous results, in this case the use of high order models takes to some advantages. In more detail the best performance are offered by the model of order three, which is the only one that achieves about the 90% of detection rate with about the 10% of false alarm rate. Moreover it is important to note that only the models of order three and four are able to correctly detect all the attacks, while the model of order one and two achieve a maximum of 97%

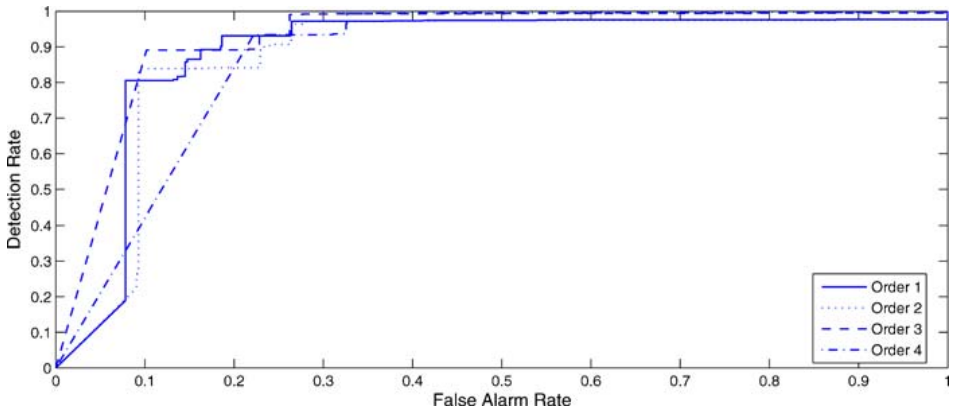


Figure 10. Performance of homogeneous Markovian models of different orders (data set 2).

DOI: 10.1002/ett

of correct detection. Finally, from the graph, we can realise that the use of a Markov chain of order four corresponds to a great worsening of the performance, when false alarm rate is 'low'. This is due to the fact that many attacks are associated with short connections, that cannot be described by using models of too high order.

Table 2 shows the detection rate achieved for each distinct type of performed attacks, by using the model of

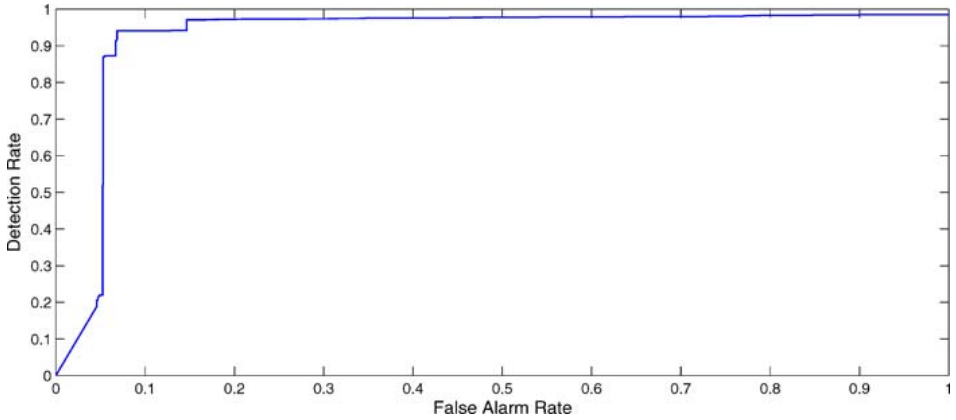


Figure 11. Performance of non-homogeneous Markov chain model (data set 2).

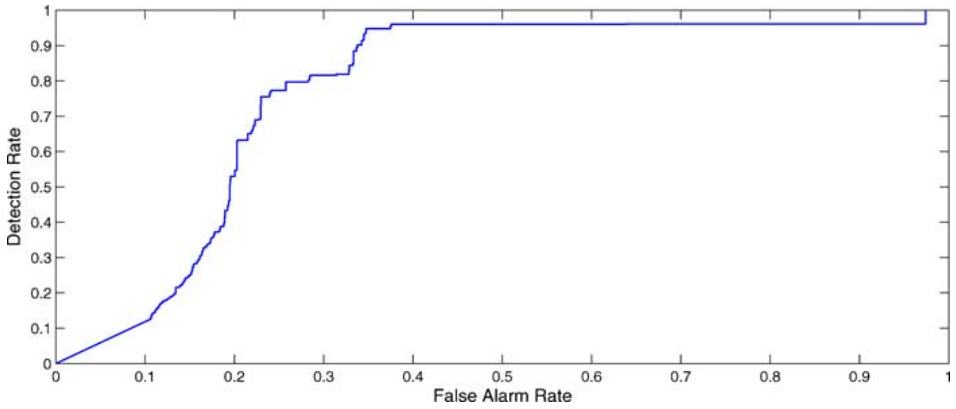


Figure 12. Performance of ECDF model (data set 2).

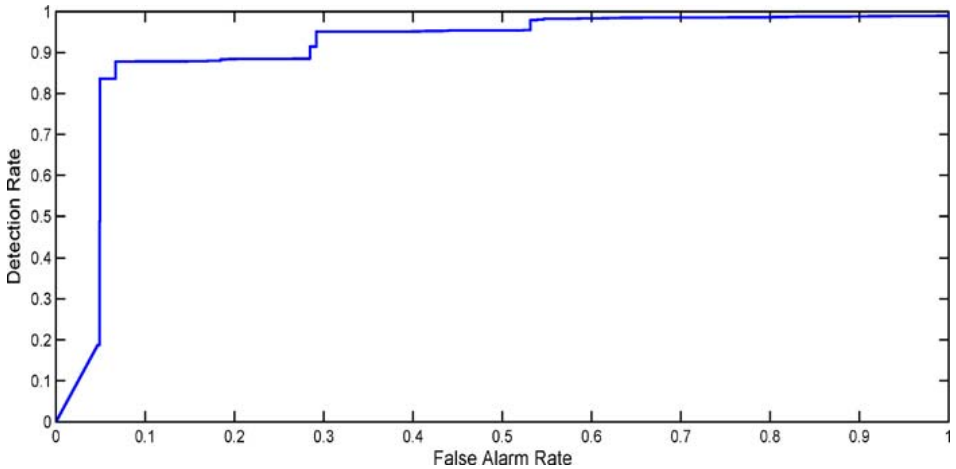


Figure 13. Performance of non-stationary ECDF model (data set 2).

DOI: 10.1002/ett

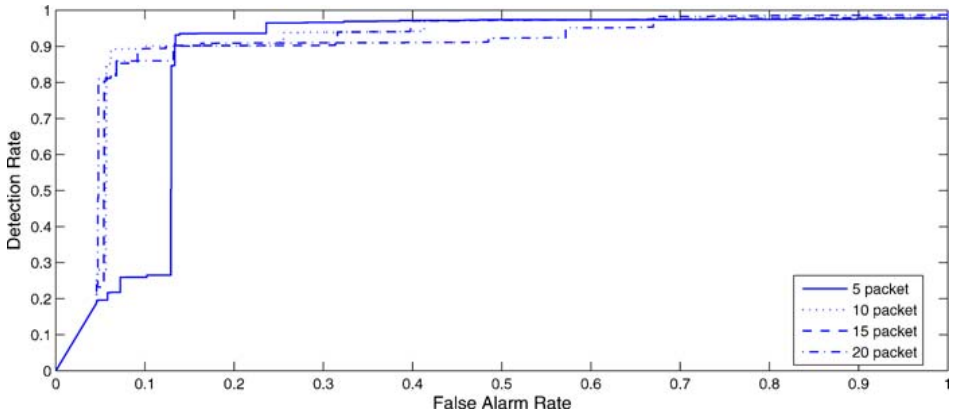


Figure 14. Performance of first order homogeneous Markov chain model, as a function of the number of processed packets (data set 2).

order three, in correspondence of a global detection rate of 90%.

The subsequent three figures show the performance of the other statistical models analysed in the paper. Figure 11 refers to non-homogeneous first order Markov chains, while Figures 12 and 13 respectively show the results achieved with stationary and non-stationary ECDF.

It is easy to conclude that ECDF is not suitable for detecting anomalies, since it achieves very bad results, while reasonable results are offered by both non-stationary ECDF and non-homogeneous Markov chain, even though they are not able to achieve a 100% of detection rate.

Finally, Figure 14 shows the performance of the first order homogeneous Markov chain model as a function of the number of analysed packets for each connection, highlighting that, as in the previous case, good performance are achieved with a small number of packets.

5. CONCLUSIONS

In this paper we have presented an anomaly based network intrusion detection system, which detects anomalies using statistical characterisations of the TCP traffic. We have considered several stochastic models, such as first order homogeneous and non-homogeneous Markov chains, high order homogeneous Markov chains and stationary and non-stationary ECDF. We have detailed the estimation of the parameters of the models. In order to compare these statistical models, we have tested our IDS over the 1999 DARPA data set, a standard benchmark for IDS performance comparison, and over 'up to date-' traffic traces, collected at the University of Pisa. The performance analysis has highlighted that the best results are obtained with the use of homogeneous Markov chains and that some improvements can be achieved using high order Markovian models. In more detail, some attacks can only be detected by Markov chains of order three and four, which are the only ones which can lead to a detection rate of 100%.

Moreover, we have shown that, since only a small quantity of packets is sufficient to reveal intrusions in the TCP traffic, this kind of approach is suitable for online detection.

ACKNOWLEDGMENTS

The authors thank Enrico Stinco and Tiziana Stoesselli for their contribution in support to this work.

REFERENCES

1. The CERT Research Annual Reports. Available at: http://www.cert.org/archive/pdf/
2. Kemmerer RA, Vigna G. Intrusion detection: a brief history and overview. *IEEE Security and Privacy (supplement to Computer Magazine)* 2002; **35**(4):27–30.
3. Stallings W. *Cryptography and Network Security*. Prentice Hall, New Jersy, USA 2007.
4. Denning DE. An intrusion detection model. *IEEE Transactions on Software Engineering* 1987; **SE-13**(2):222–232.
5. Ye N, Yebin Zhang Y, Borror CM. Robustness of the Markov-chain model for cyber-attack detection. *IEEE Transactions on Reliability* 2004; **53**(1):116–123.
6. Ju W-H, Vardi Y. A hybrid high-order Markov chain model for computer intrusion detection, NISS. *Technical Report No. 92*, February 1999.
7. Schonlau M, *et al*. Computer intrusion: detecting masquerades, NISS. *Technical Report No. 95*, March 1999.
8. Ye N, Ehiabor T, Zhanget Y. First-order versus high-order stochastic models for computer intrusion detection. *Quality and Reliability Engineering International* 2002; **18**:243–250.
9. *Tcpdump/Libpcap Public Repository.* [online] Available at: http://www.tcpdump.org

10. Raftery AE. A model for high-order Markov chains. *Journal of the Royal Statistical Society, Series B* 1985; **47**:528–539.
11. Raftery AE, Tavare S. Estimation and modelling repeated patterns in high-order Markov chains with the mixture transition distribution (MTD) model. *Journal of the Royal Statistical Society, Series C - Applied Statistics* 1994; **43**:179–200.
12. Vardi Y, Lee D. From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problem. *Journal of the Royal Statistical Society, Series B* 1993; **55**:569–612.
13. Iusem AN, Svaiter BF. A new smoothing-regularization approach for a maximum-likelihood estimation problem. *Applied Mathematics and Optimization* 1994; **29**:225–241.
14. Mood AM, Graybill FA, Boes DC.*Introduction to the Theory of Statistics* (3rd edn). McGraw-Hill: Tokyo, Japan, 1974.
15. MIT Lincoln Laboratory, DARPA Evaluation Intrusion Detection. [online] Available at: http://www.ll.mit.edu/IST/ideval/
16. Lippmann R, *et al*. The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks* 2000; **34**(4):579–595.
17. Haines JW, *et al*. 1999 DARPA intrusion detection evaluation: design and procedures. *Technical Report*, February 2001.

AUTHORS' BIOGRAPHIES

Christian Callegari received his Laurea degree in Telecommunication Engineering 'cum laude' on October 2004 from the University of Pisa, discussing a thesis titled 'Simulative analysis of RSVP-TE, and evaluation of end-to-end rerouting techniques in MPLS networks'. In 2005, he obtained the qualification to practice the profession of Engineer and he is a member of the IEEE Communication Society. On January 2005, he joined the department of Information Engineering at the University of Pisa as a Ph.D. student, and in 2008, he obtained his Ph.D. in Information Engineering. He is actually a post-doctoral fellow in the Telecommunication Network research group at the department of Information Engineering of the University of Pisa. His research and professional areas of interest are network security, traffic engineering, MPLS architecture and network simulation.

Sandrine Vaton was born in Laon, France, in 1970. She has received the Engineer's degree from the Ècole Nationale Supérieure des Télécommunications (ENST) in 1994 and the Master's degree in Applied Probabilities from the University 'Pierre et Marie Curie' in 1995. She has obtained a Doctor's degree in Signal Processing from the ENST in 1998. Since 1999 she is an associate professor at the ENST Bretagne in Brest, France, where she teaches queuing systems, cryptography and computer simulation. Her researches address digital communications, network measurements, traffic modelling and performance evaluation.

Michele Pagano received laurea (cum laude) in Electronics Engineering in 1994 and a Ph.D. in Electronics Engineering in 1998, both from the University of Pisa. Since 2007, he is an associate professor at the Dipartimento di Ingegneria dell'Informazione of the University of Pisa, where he is the official instructor of the courses of 'Telematics', 'Performance of Multimedia Networks' and 'Network Security'. His research interests are related to statistical characterisation of traffic flows and to network performance analysis, mainly in the framework of architectures able to support Quality of Service. Performance evaluation has been carried through analytical approaches as well as by means of discrete event simulation. Finally, a new research field is represented by network security issues, mainly in the framework of Intrusion Detection Systems. He has co-authored around 100 papers published in international journals and presented in leading international conferences.

DOI: 10.1002/ett

INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2010; **00**:1–24

A scalable, efficient and informative approach for anomaly-based Intrusion Detection Systems : theory and practice

Osman Salem[1], Sandrine Vaton[2] and Annie Gravey[2]

[1] *Laboratoire d'Informatique Paris Descartes (LIPADE)*
Université Paris Descartes
Paris, France
[2] *Département Informatique*
TELECOM Bretagne
Brest, France

SUMMARY

In this paper, we present the design and implementation of a new approach for anomaly detection and classification over high speed networks. This approach is efficient in terms of its scalability, the anomaly detection delays and its ability not only to detect an anomaly but also to provide detailed and accurate information about the ongoing attack (attack type, IP addresses, port numbers, etc.).

The proposed approach is based first of all on a data reduction phase through flow sampling by focusing mainly on short lived flows. The second step is then a random aggregation of some descriptors such as a number of SYN packets per flow in two different data structures called Count Min Sketch and Multi-Layer Reversible Sketch. A sequential change point detection algorithm continuously monitors the sketch cell values. An alarm is raised if a significant change is identified in cell values. We make profit of the reversibility properties of the Multi-Layer Reversible Sketch to retrieve useful information about which "flow" was the culprit one. The good properties of the Count Min Sketch in terms of resistance to collisions make it possible to check the accuracy of the information about the ongoing attack. With an appropriate definition of the combination of IP header fields that should be used to identify one flow we are able not only to detect the anomaly but also to classify the anomaly as DoS, DDoS or flash crowd, network scanning and port scanning.

We validate our framework for anomaly detection on various real world traffic traces and demonstrate the accuracy of our approach on these real-life case studies. Some of our test traffic traces are measurements obtained on a geographical and technical subdivision of an ADSL network of a major Internet Service Provider. Other traces were collected in the framework of a French national research project during online experiments on a distributed measurement network with well known attack types and instants. Our analysis results from online implementation of our algorithm over measurements gathered by a DAG sniffing card are very attractive in terms of accuracy and response time. The proposed approach is very effective in detecting and classifying anomalies, and in providing information by extracting the culprit flows with a high level of accuracy. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Network anomaly detection, Change Point Detection, Multi-chart Cumulative Sum, DoS, Sketch

1. INTRODUCTION

The daily available patches and updates for servers, and the filtering of malicious packets, offer a significant protection from known vulnerability attacks. A talented attacker can still bypass these defenses by detecting and exploiting new vulnerabilities in the latest software releases. This is a subtle attack that requires a lot of skills and efforts on the part of the attacker, and it is not very common. There are much easier ways to deny service and to silence any web service. While an experienced attacker may use botnets of tens of thousands of compromised hosts with one of the popular DDoS bots (Agobot, SDBot, RBot, SpyBot, etc.), an inexperienced attacker may rent compromised machines from available web sites. The intentions behind these attacks often differ, and range from revenge, vandalism, political reasons up to money extortion.

In DoS attacks, the target machine (or network) spends all of its critical resources (such as bandwidth, CPU time, memory, etc.) on handling the attack traffic and cannot provide service for legitimate clients. With DDoS, security threats for computer network availability have increased significantly. Recent DDoS attacks in August 2009 were performed against Twitter, Google and Facebook. Twitter was driven offline for 3 hours by the attacks, and there was degraded service (slow down) for the users of Facebook. This time the reason was political and directed against an individual who blogged about the independence of a breakaway region of Georgia. Attackers wanted to silence this blogger and they did. Flooding can easily lead to the disruption of critical infrastructure services and degrades the QoS in ISP networks. Legitimate traffic cannot find resources, and gets dropped because of the high attack volume. Therefore, an effective detection of anomalies requires the ability to separate the malicious traffic from legitimate traffic, that is to say that some additional information is necessary, for example some information about the victim servers, the attackers and the type of ongoing attack (DoS, scanning, etc.). This information is needed in order to take the appropriate countermeasures and protect the access for normal users.

Intrusion detection systems (IDS) classify network traffic based on some classification rules. They are divided into two categories: misuse based and anomaly based IDS. A misuse based IDS is based on signatures, and looks into the packets for a matching of the predefined attack signatures (e.g. Snort [45], Bro [43]). It raises an alert when a suspicious activity has been identified. Like an antivirus, misuse based systems require a regular update of their signatures database to detect recent attacks. Although these methods are very efficient in detecting known attacks, they cannot analyze the content of encrypted data. Moreover a zero day attack (security hole without available fix/signature) does not have an available signature, and consequently cannot be detected. On the other hand, anomaly based IDS identify deviations from normal traffic patterns as anomalies. These methods try to detect changes in some traffic descriptors at the flow level (e.g the number of half open connections, requests, etc.) based on some measurements of the same parameters in some past intervals. Any inconsistent deviation in the value of these parameters is considered as an anomaly. The main advantage of anomaly detection based approaches is their independence from any prior knowledge of intrusion signatures, so that such systems may be able to detect new types of attacks. Their drawback is that they need a learning phase without attacks, and that they are not able to detect attacks that do not change the traffic pattern.

The meaning of anomaly is sometimes misunderstood, since anomaly does not necessarily mean malicious, and on the other hand a malicious behavior may not provoke any visible anomaly in the traffic. For example, both DDoS and flash crowds trigger an abnormal increase in the number of SYN towards one destination, but DDoS are malicious attacks aiming at shutting down the victim server, whereas a flash crowd is caused by legitimate users requests. We consider the problem of anomaly detection over high speed links, such as SYN flooding, network & host reconnaissance, and

worm propagation leading to significant changes in some measurable network characteristics when compared to the normal behavior. These attacks send a large number of malicious requests towards the victim server/network. As the majority of attacks today are performed using TCP [41] by exploiting its handshake procedure it is primordial to detect these attacks (e.g. SYN flooding attacks) at an early stage of their occurrence, and especially before exceeding the limit of half-open connections of the victim server. It is expected that an early detection will provide sufficient time for defense reaction, such as filtering, pushback and traceback.

In this paper we design a scalable and efficient framework for anomalies detection and classification over high speed links. The proposed framework considers the online detection of some abrupt changes in some time series related to some aggregated numbers of SYN. The analysis is performed at the flow level, as flow level monitoring permits a considerable compression of the information. A flow (as defined by Netflow or IPFIX [17]) is a set of unidirectional packets sharing the same value of 5-tuple in their IP headers: Source IP (SIP), Destination IP (DIP), Source Port (SP), Destination Port (DP), and Protocol (P). To detect anomalies over high speed links, our first step is to reduce the amount of input data through flow sampling, in order to focus mainly on malicious flows. Indeed, most of the time the network traffic is normal and the operation of analyzing all the collected data is extremely resource consuming: it consumes for example bandwidth for report transmission to NOC, and CPU time for processing the data in order to identify and extract suspect flows. In this paper we describe a Network based Intrusion Detection System (NIDS) and not a Host Based Intrusion Detection System (HIDS). A record is built for each flow at measurement points. Some of these records are exported to a central Network Operation Center (NOC). The procedure of selection of the records is a stochastic sampling procedure. The aim of the sampling procedure is to focus mainly on short-lived flows, to reduce the bandwidth used for exporting the records , and to reduce the CPU load induced by records processing at NOC.

The data reduction phase is based on the observation that short-lived flows are the source of many attacks (DDoS, PortScan, NetScan, worm spreading, etc.). During our experiments and analysis of real IP traces, we indeed checked for example that many non spoofed DoS attacks carry different source port (SP) values in their TCP segment headers so that each packet generates a small flow that is observed in the monitoring process. In real life, IP address spoofing remains a security issue for the attacker, who does not want to loose the control of bots, or be caught after bots identification.

The proposed sampling procedure reduces the large amount of raw data, and aims to improve the detection of anomalies and to reduce the false alarms by focusing on short-lived flows. However, even after flow level filtering, maintaining information for each active flow in the selected subset is a cost prohibitive approach over a high speed link. The memory and CPU requirements induced by the large state space of traffic flow identifiers (e.g. IP addresses) are not adequate for a real time analysis. It is not surprising that many existing anomaly detection algorithms have been applied on time series of some aggregated counters (e.g. # SYN packets) in one measurement point [53, 54, 49, 57]. Aggregating the counters over all the flows does not permit to trace back the culprit flows and thus limits considerably the possibility of countermeasures. As this does not reveal any information about attacker/victim for mitigation the interest of deploying such monitoring infrastructure is rather limited. Moreover the application of change point detection algorithms to the aggregated traffic tends to be inaccurate in finding attacks without generating too many false alarms since many attacks induce only small variations with respect to the whole traffic volume. On the contrary our goal is to identify network anomalies such as DoS or scans in near real-time manner, and to extract the culprit flows for further mitigation.

In response to the scalability limitations of maintaining some information such as SIP, DIP, SP, DP

or some combination of these fields for each active flow, an efficient data structure based on $k - ary$ hash tables (Fig. 1), called sketch [16, 32, 37], was proposed and used to handle large state spaces, with a small amount of memory requirement and a constant computational (update/query) complexity. A sketch is a multi-stage Bloom filter based on random aggregations of counters in shared cells, where flows identifiers (denoted by keys) are hashed to index into a set of cells in different stages using k different hash functions, usually chosen to reduce collision effects, to uniformly distribute keys, and to reduce correlations between the hash functions. To use sketch in the context of network anomalies detection, IP flows are typically classified by some combination of fields in their packet header, such as destination IP address (DIP), or source and destination IP address (SIP—DIP), destination port (DP), etc. This flow identifier is a key used to update each of the hash tables by a value which is a reward associated with the key, for example a number of SYN packets. In our paper the selection of which combination of the IP header fields will define a flow is decided in such a manner that it is possible to recognize some particular types of anomalies, such as DoS, DDoS, flash crowds, alpha flows, worms, network scans or port scans. Authors in [21] have shown that a random aggregation of flow related counters in a sketch does not significantly disrupt flow variations. Moreover they introduced the Count-Min Sketch (CMS) algorithm [21], which returns an approximate value of the counter of a given key, i.e. the accumulated value for a key. In our proposed architecture we will use a multi-stage Bloom filter or sketch structure similar to the one used in the CMS algorithm.

The proposed framework is based on detection of change points in the cells of the CMS data structure, which aggregates multiple data streams from high speed links in the stretched database. To detect significant changes in the sketch cell values, we use the Multi-chart Non-Parametric CUSUM algorithm (MNP-CUSUM [51, 31]) over the time series inside all cells of the sketch. Each time series is an accumulated number of SYN packets for all flows indexing into a particular cell. The time series are monitored by a MNP-CUSUM algorithm which goal is to identify that a change has occurred in one of the cells, and also in which cell it has occurred. MNP-CUSUM was selected because of its optimality properties in terms of false alarm rates, delays, and false localization probabilities.

With an appropriate definition of which IP header fields define a “flow’ an anomaly (DoS, DDoS, port or network scanning, etc.) results in an abrupt increase of the times series in some cells of the CMS sketch. This abrupt change is detected and the cell index values are identified by the MNP-CUSUM algorithms. In order to permit the mitigation of the ongoing anomaly some identifier of the culprit flows is necessary. Therefore, an additional Multi-Layer Reversible Sketch (MLRS) is introduced and used in parallel to the CMS structure for software efficient sketch inversion, in order to extract bad flows after the detection of the anomalies.

The functional operation of the system is the following: we proceed to filter flow-record data by focusing on small flows only, and we update the counters of two compact sketches (MLRS and CMS) for discrete time interval T. Afterward, MNP-CUSUM algorithm is used to check the presence of buckets which value deviates significantly from normal behavior. After the detection of anomalies by CUSUM algorithm, we recover the key associated to cells with raised alarm by CUSUM in MLRS, through exploiting the cell index to recover responsible flows identifier, and we achieve verification through count-min query of alarm value for suspect key over the CMS sketch.

The main contribution of our approach is to combine skilfully different components into a complete framework for Intrusion Detection Systems (IDS). The proposed IDS has many attractive properties: scalability, small delay as well as low False Alarm Rate (FAR), and a rich and accurate output information (attack type and instant, IP addresses and port numbers, etc.). The complete framework inherits from the good properties of its main component blocks : the reversibility of the MLRS permits retrieving useful information about the ongoing attacks, the MNP-CUSUM makes it possible to detect

the anomalies with a low delay and a low FAR, the CMS is designed in order to avoid that collisions generate false alarms. The performance of the complete framework is extensively assessed on several real-life traffic traces including online experimentations with well-known attack types and instants.

The remainder of this paper is organized as follows. Section 2 presents some related works about anomaly detection. In Section 3, we briefly review two important building blocks in our algorithm : the CMS data structure and the MNP-CUSUM anomaly detection algorithm. Section 4 describes our complete framework for anomaly detection and classification over high speed networks. In Section 5, we present the analysis results from the application of the proposed framework over real Internet traces, and we discuss its effectiveness in terms of true detection, false alarm rate and accuracy of the output. Finally, Section 6 presents concluding remarks.

2. RELATED WORKS

Many important contributions have been proposed to undermine anomalies in network traffic. We can cite for example Haar-wavelet analysis [39], entropy based methods [42] , sequential change point detection methods with the CUSUM algorithm [49, 54, 51], adaptive threshold analysis [12], exponentially weighted moving average method (EWMA) [57], Holt-Winters seasonal forecasting based methods [11], data reduction techniques with sketches [32, 37], SNMP MIB statistical data analysis [58], Principal Component Analysis (PCA) [33, 28], etc.

When early approaches for anomaly detection were focused on the definition of models able to represent the traffic pattern, other advanced works aggregate the whole stream of packets in one time series, and apply a change point detection algorithm to detect the instant of anomaly occurrence [49, 54]. The latter have a good performance in terms of spatial and temporal complexities, but present the drawback of aggregating all the traffic in one flow, especially over high speed networks, where low intensity attacks cannot be detected with such a method. Furthermore, discovering the time instant of an attack occurrence without any additional information about the malicious source or victim is not enough to react against the attack. Usually, the amount of traffic is huge, and manual search/extraction of the malicious flows is a difficult operation. Therefore, to increase the accuracy of these methods, and to uncover the victim or attacker and classify the detected anomalies, several approaches have been proposed in the literature [37, 59, 24, 47]. However, the applicability of such on-line approaches for packet processing requires FPGA equipment [47].

Schweller *et al.* in [47, 48] propose the use of random aggregation counters for more fine grained detection. To discover the victim of flooding, they propose a method based on Galois Field $GF(2^l)$ for mangling and for simplifying sketch inversion. The proposed method is hardware efficient, and has been implemented in FPGA. Bu *et al.* in [13] propose an extension to the previous method through sequential hashing to reduce the complexity of previous sketch inversion methods. Feng *et al.* in [24] propose a method based on XOR operator and linear algebra for sketch inversion. In this paper, we will briefly show another method for reversing sketch through the use of an additional 2D table and RC4 stream ciphers.

All these proposed approaches have been used either to detect the heavy hitter flows (most frequent flows) or to detect an abrupt deviation between two discrete intervals via a simple comparison. Many different methods have been used in order to uncover anomalies in traffic flows. In-house methods do not have optimality properties and suffer from many shortcomings such as false alarms, instabilities, sensitivity to the training period, etc. [44]. In this paper we will use the multi-chart non parametric CUSUM algorithm [51] over sketch in order to uncover changes. Indeed the optimality property of

CUSUM algorithms is translated in practice into less false alarms and smaller detection delays than in-house methods. In this paper, we will apply the recursive CUSUM algorithm over sketch.

We are not the only one who detect anomalies in traffic with CUSUM based approaches. Indeed the CUSUM algorithm has been used in order to detect a variety of different security problems (mainly DoS/DDoS and worms) from traffic inspection. For example, Wang et al. (2002, 2004) [54, 53, 55] detect SYN flooding and DDoS attacks. Wang *et al.* in [54] aggregate the whole traffic in one flow, and use a non parametric version of CUSUM for detecting TCP SYN flooding. They consider different metrics such as number of SYN, FIN and SYN/ACK in CUSUM for detecting flooding attacks. Siris *et al.* in [49] evaluate and compare two anomaly detection algorithms (adaptive threshold and CUSUM) for the detection of TCP SYN flooding. The result of the comparison shows that CUSUM is more efficient for detecting low intensity attacks than adaptive threshold. Lim et al. (2005) [38] implement SYN flooding detection methods on a programmable network processor. He et al. (2008) [26] focus on available bandwith estimation and DDoS detection. As the vulnerabilities of wireless communication protocols are the vectors of many attacks today, some papers use in their detection mechanism the CUSUM algorithm : Lee et al., (2007) [35] detect DoS attacks on 3G wireless networks ; Yan et al. (2009) [56] detect Bluetooth worm propagation with CUSUM and GLR. CUSUM has also been used by several authors in order to detect worm propagation : Bo et al. (2005) [9] detect worm attacks, Chen et al. (2006) focus on worm outbreaks and SYN flooding, Bu et al. (2006) [14] detect scan-based worms in darknets. Darknets traffic has been studied by different authors : Ahmed et al., (2009), [4, 5] detect nested anomalies in darknet traffic with a sliding-window mechanism and CUSUM. Tartakovsky et al. (2007) have introduced the now celebrated multi-chart CUSUM [51] and initially applied it to the monitoring of production networks (packet size analysis). The same author (Tartakovsky et al., 2006, [50]) has compared the performance of decentralized distributed change detection methods. Kang et al. (2009) study botnet detection with entropy based multi-chart CUSUM [29]. As attacks against encrypted protocols such as SSH and SSL evade signature-based IDS, statistical approaches such as CUSUM are particularly useful in that case ; Fadlullah et al. (2007) [23] detect attacks against ciphered protocols with CUSUM.

As the main problem is the huge amount of data to process, many researchers have tried to reduce the size of collected data before processing, through packet filtering and sampling. The most famous example is NetFlow [18] in CISCO IOS, which can be configured with uniform packet sampling. Packet sampling has been widely studied, some works have investigated how packet sampling impacts the precision of anomaly detection algorithms [10, 40], and others propose various sampling techniques to improve accuracy in anomaly detection.

As many packet sampling methods change the characteristic features of traffic flows, it is easy to miss flows with a small number of packets, and to incorrectly estimate the flow size, especially during the identification of mice and elephant flows. In general, if the flow is built from sampled packets, the precision of the anomaly detection system depends on the sampling rate. Kawahara *et al.* [30] show that packet sampling degrades the results of anomaly detection algorithms. Hohn *et al.* [27] compared packet sampling with flow sampling and found that flow sampling outperforms packets sampling in recovering flow distributions. Mai *et al.* [40] present the impact of random packet sampling and random flow sampling on anomaly detection. Results revealed that both degrade anomaly detection. However, random flow-sampling outperforms random packet sampling, because packet sampling introduces a bias that degrades the detection effectiveness and increases the number of false alarms.

Recently, Androulidakis *el al.* in [7, 6] designed and analyzed selective flow sampling in order to improve the accuracy of anomaly detection algorithms. They investigate the impact of sampling on the performance of non parametric CUSUM and entropy based anomaly detection, and they prove that

selective flow sampling achieves ”magnification” of the anomalies. This sampling method has inspired the sampling algorithm used in this paper.

The approach presented in this paper uses sampling to reduce the amount of data and to discard unpredictable variations of legitimate traffic. Afterward, it uses the sequential MNP-CUSUM over sketch for anomaly detection thus allowing us to detect changes with a small delay a low false alarm rate. A new software efficient approach for sketch inversion through encryption and index exploitation is proposed to provide information about victim/attacker. An appropriate definition of which IP header fields define a flow makes it possible to classify the anomalies by categories (DoS, DDoS, network or port scanning, etc.).

After the seminal papers by Cormode and Muthukrishnan [19, 20, 21] a few methods which combine the use of sketches and time series analysis methods have been published [34, 46, 22, 15]. Lakhina *et al.* in [34] improve the performance of their Principal Component Analysis (PCA) based system by inserting the use of sketch structures. Analysis of PCA methods performance in the discovery of traffic anomalies have revealed that these approaches are sometimes prone to false alarms and misdetections [44]. Indeed PCA methods are data-driven and require a particularly long anomaly free training phase in order to calibrate the decomposition basis. This leads to a risk of incorporating some of the anomalies of the training dataset into the decomposition basis. Some authors combine the use of sketch and a multiresolution (or multiscale) analysis based on wavelets in order to undermine traffic anomalies [22, 15]. Wavelet analysis is used to detect discontinuities (irregular patterns) in traffic data. One of the advantages of wavelet analysis is that it does not need a training phase with anomaly free traffic. Dewaele *et al.* analyze a 6 years long traffic trace on a trans-Pacific backbone link (MAWI dataset). They discover a large number of irregular patterns; some are due to known anomalies but the method also reveals a large number of unexpected flows, be their nature legitimate or not remaining still an open issue. Example of ”anomalous” flows can be elephant flows (HTTP traffic, FTP or SSH connections), or destination IP addresses receiving small number of packets from a large number of source IP addresses(P2P traffic). The key used in order to index the cells in the sketch data structure in [22, 15] is a destination IP address (DIP) or a source IP address (SIP). In [36] the authors also detect changes in high-dimensional traffic data. They use a non parametric change detection test based on the U statistics. Two different dimension reduction techniques are used: TopRank which is based on record filtering and HashRank which is based on random aggregation.

Our method combines sketch data structures (CMS and reversible sketch) and the MNP-CUSUM to discover in real-time significant changes in the number of SYN packets associated to some ”flows”. Flows are associated to some particular fields in the IP and TCP headers : source IP address (SIP), destination IP address (DIP), source port (SP), destination port (DP) or a combination of those fields. There are a number of advantages to using our algorithm compared to other published algorithms. MNP-CUSUM algorithm is particularly simple and computationally efficient compared to decomposition methods such as PCA or wavelet analysis. Moreover CUSUM is sequential by nature which makes it adapted to on-line treatments. The optimality properties of the CUSUM have been theoretically established (low false alarm rate, low delay) and this translates into practice into fewer raised false alarms than decomposition based methods. We recognize that MNP-CUSUM is less sensitive to subtle irregularities in traffic than wavelet based analysis and thus reveals less ”anomalies” in traffic than [22]. Using a non-parametric version of the CUSUM algorithm (NP-CUSUM) makes the algorithm robust against non stationarities in traffic. Another very important feature of our algorithm is that it reveals precise information about the ongoing attack. Many methods raise alarms but then a manual inspection of the traffic is often necessary in order to analyze the traffic and to classify the anomaly (DoS, scan, false alarm, etc.) Our algorithm automates the classification step by using

appropriate combination of TCP/IP header fields as keys to index cells in the sketch data structures. Our algorithm moreover pinpoints malicious flows as it includes a sketch reversion step in order to recover the value of the keys from the index of anomalous cells.

3. THEORETICAL BACKGROUND

In this section, we briefly survey the underlying Count-Min Sketch (CMS) data structure and Multi-chart Non-Parametric CUSUM (MNP-CUSUM) used in our framework.

3.1. Count-Min Sketch

Cormode and Muthukrishnan [21] introduced another kind of multi-stage Bloom filter called the Count Min Sketch (CMS). The advantage of the proposed algorithm is to provide an estimate of the associated counter with a key in the stretched data structure (CMS). Let $S = s_1 s_2 \dots s_n$ be the set of input stream, where each item $s_i = (\kappa_i, \nu_i)$ is identified by a key $\kappa_i \in U$, drawn from a fixed universe of items U. $\nu_i \in \mathbb{R}$ is the reward associated with each key. For example, with $\nu_i = \#SYN$ and $\kappa_i =$ DIP, the goal will be to count the number of SYN packets corresponding to the different destination IP addresses. The sketch data structure is made up of d hash tables. The arrival of an item with key κ_i increments its associated counter in the j^{th} hash table by ν_i ($C_{j,h_j(\kappa_i)} += \nu_i$), as shown in Figure 1. The update procedure is realized by d different hash functions, chosen from the set of 2-universal hash functions $h_j(\kappa_i) = \{((a_j \kappa_i + b_j) \bmod P_U) \bmod w'\}$, to uniformly distribute κ_i over hash tables and to reduce collisions. The parameter P_U is a prime number larger than the maximum number in the universe, where Mersenne prime numbers of the form $2^i - 1$ are generally chosen for fast implementation. a_j and b_j are random integers smaller than P_U, with $a_j \neq 0$. To highlight the use of 2-universal hash function in the context of IP address ($P_U = 2^{61} - 1$) let us take κ_i equal to 1.2.3.4, with random values of $a_j = 2$ and $b_j = 3$, and a sketch width $w' = 256$, the result of hash function is $h(1.2.3.4) = 4$. This means that the associate bucket in the first line of the sketch is the number 4.

(a) Sketch structure. (b) Update with $\kappa_i = 1.2.3.4$ & $\nu_i = 5$.

Figure 1. Sketch data structure.

The Count-Min point query returns an estimate of the counter for a given key κ_i as the minimum of the corresponding d counter values : $\hat{s}_k(\kappa_i) = \min_{0 \leq j < d} \{C[j][h_j(\kappa_i)]\}$.

The definition of the accuracy of the result of a CMS query is probabilistic. The distance between the estimated and true values of the count should be lower than a precision ϵ with an error probability at most $\delta : \mathbb{P}(d(\hat{s}, s) \leq \epsilon) \geq 1 - \delta$. With this constraint, the parameters d (number of pairwise independent universal hash functions) and w' (number of hash values) should be chosen as follows : $d = \lceil \ln(1/\delta) \rceil$ and $w' = \lceil e/\varepsilon \rceil$ where e is the base of the Neperian logarithm. Thus, it maintains modest storage

requirements of $\ln(1/\delta) \times (1/\varepsilon)$ count cells.

Flow records from collected data traces, can be classified into series of (κ_i, v_i), where κ_i can be the concatenation of DIP and DP ($DIP|DP$), or any other combination from flow record identifiers, and the value ν_i can be, for example, the number of SYN requests. The CMS query can check for example *if* a given $DIP|DP$ is under SYN flooding attack by verifying the value of $\hat{s}_k(\kappa_i)$. But it is unable to identify *which* $DIP|DP$ is under attack.

3.2. MNP-CUSUM

To uncover anomalies we use the multi-chart non parametric CUSUM algorithm (MNP-CUSUM, [51]). The CUSUM algorithm is a sequential change point detection method. Suppose that a flow of data is monitored sequentially : $y_1, y_2, y_3, \ldots$. The goal of a sequential change point detection algorithm is to detect with a delay as small as possible a change in the distribution of the data y_i. Suppose that the pre-change and post-change distributions are known and that they are characterized by probability density functions (p.d.f.) $f_{\theta_1}(y_k)$ and $f_{\theta_2}(y_k)$. The alarm time in the parametric version of the CUSUM algorithm is defined as $t_a = \min\{k \geq 1 : g_k \geq h\}$ where h is a threshold and g_k is the test statistics which can be computed sequentially : $g_0 = 0, g_k = \max(0, g_{k-1} + \log \frac{f_{\theta_2}(y_k)}{f_{\theta_1}(y_k)})$. The rational behind the CUSUM algorithm is that before the change the quantity $\log \frac{f_{\theta_2}(y_k)}{f_{\theta_1}(y_k)}$ is on the average negative, whereas after the change it is on the average positive : as a consequence, the test statistics g_k remains around 0 before the change, and it increases linearly with a positive slope after the change, until it reaches the threshold h when the alarm is raised. This is illustrated on Figure 2. A change in the mean of a Gaussian time series is detected. S_k is the cumulated log-likelihood ratio which is defined as $S_0 = 0, S_k = S_{k-1} + \log \frac{f_{\theta_2}(y_k)}{f_{\theta_1}(y_k)}$. $s_k = \log \frac{f_{\theta_2}(y_k)}{f_{\theta_1}(y_k)}$ is the log-likelihood ratio. One can observe from this figure that S_k has a negative slope before the change point and a positive slope after the change has occurred.

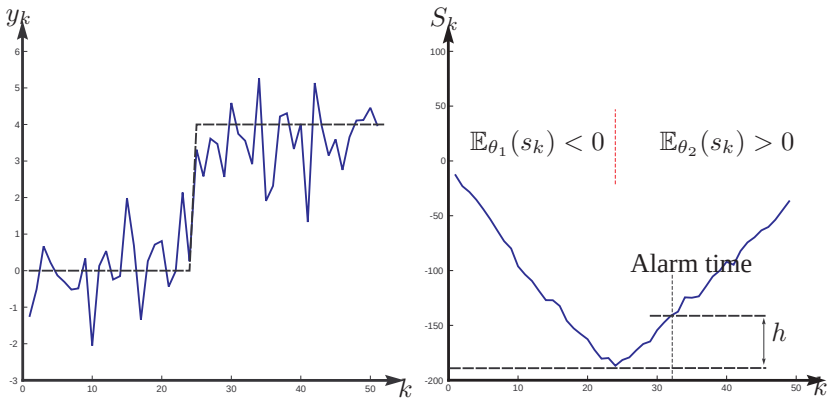


Figure 2. Intuitive derivation of the CUSUM.

The CUSUM algorithm is asymptotically optimal in the class K_γ of tests with average time between false alarms bounded by γ. More precisely, it reaches the Lorden bound which states that among the class K_γ the average detection delay is at best proportional to $\log(\gamma)$ as $\gamma \to \infty$ with a

proportionality factor that depends on the Küllback-Leibler divergence between pre-change and post-change distributions.

Now suppose that we have to monitor several channels or sensors jointly : for example, $y_k(i)$ will be the number of SYN packets during a time interval k for a subset i of all DIP adresses (e.g. those who collide in the same cell number i of a given hash table). Until the unknown change time t_0 each random value follows a distribution $f_{\theta_1,i}(y_k(i))$ and after t_0 a change occurs in the distribution $f_{\theta,i}(y_k(i))$ of one channel only, say channel number i. The goal is to identify with a small delay that the change has occurred and also in which channel it has occurred. The multi-chart parametric CUSUM [51], simply called LR-CUSUM, is defined as : $t_a = \min_{1\leq i\leq N} t_a(i)$ where $t_a(i) = \min\{k \geq 1 : g_k(i) \geq h_i\}$. h_i is a threshold adapted to the channel i (constant in our case) and $g_k(i)$ is a CUSUM test statistics for channel number i : $g_0(i) = 0, g_k = \max(0, g_{k-1}(i) + \log\frac{f_{\theta_2,i}(y_k(i))}{f_{\theta_1,i}(y_k(i))})$. The alarm is raised when one of the test statistics $g_k(i)$ reaches the threshold h_i and the change is declared in the channel i. The LR-CUSUM is asymptotically optimal in the sense that (i) the average detection delay for a change of type i is asymptotically linear with h (when $h \to \infty$) with a slope that is related to the entropy of the pre-change distribution, and (ii) if $h = \log(N\gamma)$ then the mean time before false alarms is greater than γ and the worst case of the average detection delay is asymptotically linear with $\log(\gamma)$ (when $\gamma \to \infty$).

Due to large variations in traffic patterns, to non stationarities, and to lack of consensus on network traffic characteristics, we cannot assume that the monitored variables follow a specific distribution. Therefore, we use the non-parametric version of the multi-chart CUSUM, as it only requires a very loose information on the distribution of the traffic time series before and after the change. In the non parametric multi-chart CUSUM (MNP-CUSUM) the log-likelihood ratio $\log\frac{f_{\theta_2,i}(y_k(i))}{f_{\theta_1,i}(y_k(i))}$ is replaced with some function $L_i(y_k(i))$ which is chosen in such a way that its average value $\mathbb{E}(L_i(y_k(i)))$ is negative before the change and positive after the change. For example, in our simulations we chose $L_i(y_k(i)) = y_k(i) - (\mu_i + c\sigma_i)$, where μ_i and σ_i represent the mean and the variance respectively. With this definition the non parametric CUSUM is sensitive to a change in the mean value of the time series, supposing that μ_i is the average value before change, and supposing that the average value of the time series after the change is greater than $\mu_i + c\sigma_i$.

Network traffic is naturally variable. Without any anomaly (attacks, flashcrowds, etc.) the traffic is subject to natural variations due, for example, to day/night effects. These variations occur on a significantly longer time scale than anomalies which can then be identified as abrupt changes in traffic patterns. In this paper, we will use CUSUM algorithm to detect short term anomalies, and the detection of longer-term anomalies (several minutes or hours) is not addressed in this paper. Because of natural variability the parameters of traffic distribution in normal operation, for example μ_i and σ_i, slowly vary along the time. In order to follow the slow trends of traffic parameters on the long term the mean μ_i and standard deviation σ_i before change are estimated recursively using the EWMA (Exponentially Weighted Moving Average) : $\hat{\mu}_i(k) = \alpha\hat{\mu}_i(k-1) + (1-\alpha)y_k(i)$ and $\hat{\sigma}_i^2(k) = \alpha\hat{\sigma}_i^2(k-1) + (1-\alpha)(y_k(i) - \hat{\mu}_i(k))^2$. The estimation of mean and variance using the EWMA can deal with seasonal variation (working hours, night, day of week, etc.), by slowly updating these statistical parameters ($\hat{\mu}_i$,$\hat{\sigma}_i^2$). It is worth noting that the algorithm is not much sensitive to the value of the weighting parameter α that should be taken close to 1 ; a typical value is $\alpha = 0.9$.

4. PROPOSED APPROACH

To detect network anomalies over high speed networks, the first logical step is data reduction and aggregation of the huge amount of collected flow records. This step is useful for several reasons, including saving the bandwidth used for report transmission, and reducing the memory requirement and the processing complexity of analyzing collected data. We achieve this through flow level sampling, by selecting small size flows (size is the number of packets in the flow), as done in [7, 6].

Flows with a small number of packets are the source of many anomalies. In DDoS/DoS, the attacker uses a spoofed IP address with each packet to evade detection, identification, filtering and tracing back. Even when using botnets, the disclosure of zombies is not in the interest of the attacker. Firstly, the identification of the attacker may be more easily discerned, and secondly he wants to keep his remote control of the zombie for future use. Therefore, spoofed source IP addresses are used by bots in each packet when launching DDoS.

In the case of NetScan, worms usually scan networks to infect new vulnerable hosts as quickly as possible (e.g. Code Red, Slammer). Network scan also allows an attacker to identify an active host for security assessment, where all exploitations are always preceded by a reconnaissance phase. In security assessment, the malicious user scans port for enumerating services to identify vulnerable applications. Each packet in all these attacks results in a new flow.

To select all small flows with size $x < B$, we choose a sampling rate inversely proportional to the number of packets in the flow ($p = 1/x$) *if* ($x \geq B$) and no sampling *if* ($x < B$). B is a threshold for the number of packets. This is a slight modification of selective sampling ($p = B/n.x$) suggested in [7]. Despite the sampling, the reduction gain ratio was approximately 5% over the used traces. Manual inspection of these IP traces with P2P traffic, shows that the majority of TCP flows have less than 3 packets/flow. In fact, the dominant number of records are small flows. Therefore, to achieve a reduction ratio of more than $k\%$, we use the random sampling technique to reduce the number of records with small numbers of packets, as shown in eq. 1:

$$p(x) = \begin{cases} rand[1,k] == 1\ ?\ 1\ :\ 0 & \textit{if } x \leq B \\ rand[1,x] == 1\ ?\ 1\ :\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

This means that the sampling rate is ($p = 1/k$) *if* ($x \leq B$) and ($p = 1/x$) *if* ($x > B$). In our simulations we took $k = 2$ and $B = 3$. At monitoring points each flow generates a record. Once the flow has expired (either because of flow termination or because of timeout expiration) the number x of packets in the flow determines the value of the probability $p(x)$. The record is then exported to the central NOC with probability $p(x)$.

It is worth noting that per flow record collection is performed at monitoring points and that records processing for anomalies detection and malicious flows identification is performed at the central NOC. Monitoring points can be for example routers in the infrastructure of the ISP. In this paper, we assume the monitoring infrastructure (monitoring points, and central collector) has always enough computing resources to monitor and analyze the attack, while the consumption of communication resources are mitigated by sampling. The sampling procedure is beneficial since it reduces the number of exported flows which results in bandwidth usage reduction for flow exports and in CPU usage reduction at NOC. Under attack situations, load shedding mechanisms, such as proposed in [8], might be considered as extensions, but these are left out the scope of the current work.

When designing a traffic monitoring system such (IDS, application recognition system, etc.) it is important to consider how easy it can be for an adversary to evade the detector. One could imagine that

the attacker would try to evade the detector by generating artificially large flows. This strategy would not succeed in the context of SYN flooding attacks for some reasons that we are going to explain. Let us assume that the attacker sends a SYN packet and goes on transmitting data without acknowledging the SYNACK packet. Then the server will send a RST packet and the TCP connection will be closed without having generating a "large flow". In fact, the attacker cancels the SYN flooding attack by generating packets in the same flow without sending SYNACK. On the other hand if the attacker sends several SYN packets in the same TCP connection these packets will be considered as duplicate SYN packets by the server and this will not result in a DoS attack. It is important to note, that small flow sampling is applicable for TCP SYN flooding and SYN reconnaissance attacks, and can not be applied to detect other attacks based on UDP or ICMP.

After data reduction, the anomaly detection phase uses random aggregation, to avoid aggregating the whole set of records into one, and to reduce the required memory in per-record treatment. The proposed approach is based on two data structures: Count-Min Sketch (CMS) and Multi-Layer reversible Sketch (MLRS) as shown in Figure 3. The theoretical background on the CMS data structure and on the MNP-CUSUM algorithm that operates over this structure is summarized in section 3. The shared counters are continuously updated from the input data stream. κ_i is a key that identifies a flow (e.g. SIP—DIP) and ν_i is a reward associated to that flow ; in our case ν_i is a number of SYN packets for this particular flow during a fixed duration period T (e.g. $T = 1$ min.) The cells in the sketch are continuously updated that is to say that for each new SYN packet the d corresponding cells in the CMS sketch are incremented by 1. d instances of the multi-channel non parametric CUSUM (MNP-CUSUM) algorithm run in the background in order to monitor each row of the sketch. The aim is to detect a significant change in one of the cells of each row. At the end of each period T the MNP-CUSUM statistics $g_k(i,j)$ are updated as follows : $g_k(i,j) = max(0, g_{k-1}(i,j) + \Delta x_{i,j} - (\hat{\mu}_{ij} + c\hat{\sigma}_{ij}))$. $\Delta x_{i,j}$ is the difference between the value of the cell (i,j) in two consecutive intervals or, equivalently, the increment of cell (i,j) during that interval. $\hat{\mu}_{ij}$ and $\hat{\sigma}^2_{ij}$ are the sample mean and sample variance of the time series of cell increments $\Delta x_{i,j}(k), k = 1, 2, \ldots$; they are computed as an exponential moving average as it was explained in section 3. In our implementation, each cell in the 2D table becomes a data structure, containing the current and previous value of the number of SYN, the sample mean $\hat{\mu}_{ij}$ and variance $\hat{\sigma}^2_{ij}$ and the value of MNP-CUSUM statistics $g_k(i,j)$. Once the statistics $g_k(i,j)$ exceeds the threshold h for one of the cells, say cell j, then the MNP-CUSUM that monitors row i raises an alarm. When we have at least one alarm in all rows of the sketch, then a global alarm is raised.

As we want to uncover culprit flows, e.g. the flows responsible for the CUSUM raised alarms, one solution for verification could be to keep the key values inside a file/database, and to re-hash these datas to find the key that maps to cells with raised alarms. This procedure is heavy in terms of storage space and update speed as it requires storing all keys for verification.

In fact, due to random aggregation and collision occurrences with hash functions, reversing sketch is a difficult operation. However, some interesting works have been proposed in [47, 13, 24] to reverse hashing, in order to identify keys associated with cells having raised alarms. The first approach [47] is based on modular hashing and mangling via Galois Field $GF(2^n)$ operators, which is complex and more efficient for hardware implementation, as it was done with FPGA equipment in [47]. The second approach [13] is an extension of the previous method. The third approach [24] is based on nonsingular matrix on $GF(\{0,1\}^n, \oplus, .)$, and requires more memory and update cost than the previous method.

Our approach to reverse sketch is based on the idea of exploiting cell index to store keys. An additional 2D table, so-called Multi-Layer Reversible Sketch (MLRS), also containing shared counters is used (as shown by the first table in Figure 3). The key is implicitly stored in this 2D table. Firstly, the key in binary is divided into l equal parts, where each part is used as index of the shared counter in

each line of the MLRS. The width of each line in the MLRS table is given by $w = 2^P$, and the number of lines is $l = \lceil N/P \rceil$. N is the number of bits used to represent the largest number in the universe of key, and P is the number of bits in each part of the key. The update procedure of the CMS and MLRS is summarized by Algorithm 1.

Algorithm 1 Sketches Update procedure

1: $Ckey = Encrypt\_RC4(key)$;
2: **for** $i = 0$ to $d - 1$ **do**
3: $\quad j = univ\_hash_i(Ckey)$;
4: $\quad CMS[i][j].counter+ = \nu_i$;
5: **end for**
6: **for** $i = 0$ to $l - 1$ **do**
7: $\quad MLRS[i][Ckey \ \& \ (2^P - 1)].counter+ = \nu_i$;
8: $\quad Ckey >>= P$;
9: **end for**

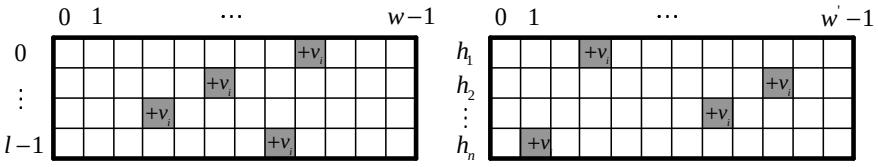


Figure 3. Multi-Layer Reversible Sketch *MLRS* and Count-Min Sketch *CMS*.

A MNP-CUSUM runs in the background for each row of the MLRS structure, as it is the case for the CMS structure. At the end of each interval T, we release a hierarchical search in each layer of the MLRS for cells with alarm raised by the MNP-CUSUM. We must have at least one cell in each layer with a CUSUM raised alarm. Otherwise, no needs to continue searching in other layers or to look in the result of the CMS.

In the simple case, when we have one alarm in each layer, the key can be recovered by concatenating the index of the l cells in MLRS. We cannot be sure of the suspect key before verification, since because of collisions with other key prefixes, the corresponding counter value can become large without any attack going on. The suspect key is verified through hashing and verifying if an alarm was raised for the corresponding cells of the CMS.

In the case of $key = DIP$, even with different widths for the decomposition procedure (8bit, 10bit, 12bit, etc.), many cells in different layers will be subject to collision occurrence (same prefix, etc.), and in some cases, we will end up with a larger set of keys to verify through *CMS* than the original key list since all possible combinations must be considered. Nevertheless, it is important to note that even if the set of suspect keys is larger than the departure one, it requires smaller memory and has a fast update time with respect to maintaining a database/file of the original key list.

To resolve this problem and reduce collision in MLRS, we use encryption with stream cipher RC4 (Ron's Code [25]) algorithm rather than Galois Field $GF(2^n)$. In cryptography RC4 is the most popular stream cipher. It is used in popular protocols such as SSL or WEP. RC4 is remarkable for its simplicity and speed in software. The used C implementation of the RC4 code is available from [3]. The RC4 algorithm is ideal for software implementation, as it requires only byte manipulations

and its implementation is based on a few lines of code. It has been proven to be powerful in our experimentations for mangling and destroying any correlation between keys having some portion in common (prefixes). The Hamming distance between cipehered keys is large even if there is some correlation between the original ones.

Encryption is a bijective function, which transforms clear text key into cipher text, denoted by $Ckey$, where $Ckey = E_S(key)$ and $key = D_S(Ckey)$. S is the shared key, and the function $E()$ must be chosen in a way to destroy any correlation between clear texts with for example the same prefix. This principle is shown in Table I for the key built from the concatenation of DIP with DP, with the use of RC4.

DIP:DP	Encrypted
192.168.100.101:80	6E96A9468CF5
192.168.100.102:80	DD08C66271E4
192.168.100.103:80	2F0F5EB19313
192.168.100.104:80	3FE7204B0435
...	...

Table I. Encryption of *DIP—DP* by RC4.

Any bijective function able to destroy correlation between keys, and return a completely random set of keys, can be used. Afterward, $Ckey$ is divided into l parts and used as an index in MLRS. To recover the key from cells with raised alarms in MLRS, we must concatenate the raised alarm indexes in each layer to get $Ckey$, and use $D_S(Ckey)$ to recover the suspect key κ_i from MLRS. This key is used for verification through CMS to ensure that all associated cells in the d hash tables have raised alarms. The hierarchical search procedure, as well as verification and sketch inversion are summarized in Algorithm 2, for a universe of size 2^N, and a width of 2^P for MLRS, $P = N/2$ and $l = 2$. A boolean alarm variable is used to indicate if the MNP-CUSUM algorithm has raised an alarm for the considered cell.

Algorithm 2 Hierarchical search and verification

```
1: for i = 0 to 2^P − 1 do
2:   if (MLRS[0][i].Alarm) then
3:     for j = 0 to 2^P − 1 do
4:       if (MLRS[1][j].Alarm) then
5:         Ckey = (j << P) | i;
6:         Alarm = cms_alarm_query(CMS, Ckey);
7:         if (Alarm) then
8:           key = decrypt_RC4(Ckey);
9:           output(key)
10:        end if
11:      end if
12:    end for
13:  end if
14: end for
```

The proposed framework can be applied to detect different types of attacks, e.g. TCP SYN flooding,

UDP packet storms, TCP/UDP PortScan, NetScan, Smurf, etc. Nevertheless, in this paper, we will only focus on TCP traffic and especially on the number of connection requests (SYN).The proposed method does not only perform anomaly detection,that is to say to raise an alarm when an anomaly is detected. It is also able to able to identify malicious flows through the use of associated key values, and to classify the anomaly by using different key definitions.

To classify anomaly, we extract from each flow record three keys ($key_1 = DIP|DP$, $key_2 = SIP|DIP$, $key_3 = SIP|DP$) through the concatenation of the binary value of two fields from each entry. These keys are used to update three instances of the proposed approach with the observed number of SYN packets with the corresponding key value for each discrete time interval (say every minute). We denote by F_i the framework instance associated with key_i. The classification algorithm can be described as follows :

Step 1. We seek to detect victims of DoS/DDoS SYN flooding. We update the counters of F_1 with the key_1 during predefined T time intervals, and we output the list L_1 of all victim servers $DIP|DP$.

Step 2. The key_2 is used to update a second instance F_2. Outputs of this step are malicious SIP, which try to scan the ports of a given DIP, if the latter is not a victim of DDoS/DoS. In contrast, if DIP is in list L_1 (i.e. victim of flooding), we store a list of suspect (LoS) whose elements are (SIP, DIP, DP), because SIP are suspects of contribution in DDoS/DoS through a static source address.

Step 3. The key_3 is used to update a third instance F_3, where output keys are SIP trying to perform a NetScan activity, if the SIP does not belong to the list LoS. Otherwise, it is the source of DDoS/DoS flooding.

The preceding three steps are used in our implementation to early identify three types of anomalies (DDoS/DoS victim, NetScan and PortScan), and provide useful information about victim or attacker. The identification of scanning attacks is based on source addresses generating too many connection requests at different ports/hosts. The PortScan and NetScan were chosen for their association with malicious attacks and worms. PortScan is often used by attackers for vulnerability assessment of running applications at victim host. NetScan are usually performed by worms in their spreading phase (random scan in code Red, linear in Blaster, bias in code Red II and Nimda, etc.) to gain access to new machines and infect them. Our proposed approach is able to detect all these kinds of scan activities.

5. EXPERIMENTS RESULTS

In this section, we evaluate the performance of using MNP-CUSUM over a Multi-Layer Reversible Sketch (MLRS) and a Count Min Sketch (CMS) structures in the detection and classification of attacks (SYN flooding attack, NetScan and PortScan). We have implemented MNP-CUSUM over sketch in C by extending the code of CMS available from [1]. We have tested the proposed algorithm over many public traces (Abilene, Auckland-II, etc.) available from [2], and other IP traces used in the OSCAR project funded by the french National Research Agency. The so-called OTIP and ADSL traces were collected and provided to the OSCAR consortium by France Telecom (FT). The ADSL trace was collected on a geographical and technical subdivision of an ADSL network. The OTIP trace is a 6.9 Gb trace made of Netflow records from CISCO routers in a FT backbone network. These traces were used as a benchmark and have been widely analyzed in the project. We also use one of the traces collected during online experiments, which were carried out to test the detection performance of the algorithms designed during the project. The topology of the measurement overlay deployed during the OSCAR project is displayed on Figure 4.

Each partner is equipped with a DAG card which sniffes all the traffic between a laboratory network

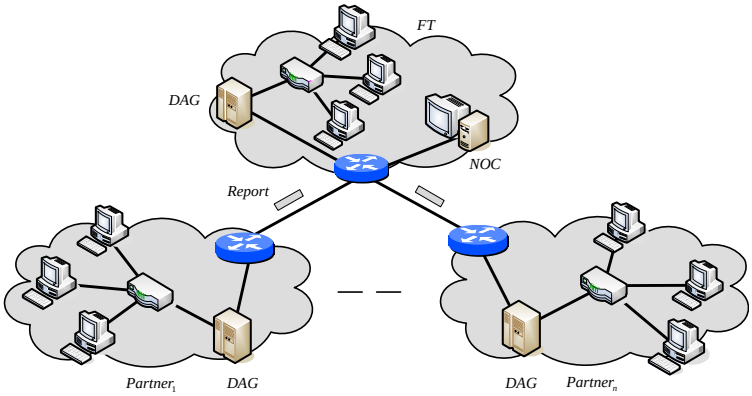


Figure 4. Topology of the measurement overlay

and a central router in the partner institution. Flow level reports are collected at the sniffing point by each partner and sent to a central Network Operation Center (NOC) by UDP sockets. Different algorithms run at the NOC in order to detect anomalies in the aggregated reports. Figure 4 describes a centralized detection architecture. Semi-decentralized architectures with anomaly detection algorithms running at the different monitoring points have also been implemented in OSCAR.

In order to reduce the spatial and temporal complexity of the proposed algorithms, OSCAR partners decided to enhance the capture process of high speed sniffing card (Endace DAG card), by adding a small C program to transform captured packets over one minute into OSCARFIX flows.

Therefore, we keep approximately the same definition of flow as the one used by Netflow [18] in Cisco routers or standardized by the IETF in the IPFIX protocol [17]. An OSCARFIX flow is defined as a unidirectional stream of packets that share the same five tuples (source IP, destination IP, protocol, source port, destination port). When a flow is considered as finished (through flow aging or TCP connection termination) a flow record is exported. The OSCARFIX flow record contains a variety of information such as the source and destination IP, source and destination ports. Instead of including the binary XOR of all TCP flags of the flow in the record as it is the case in Netflow, OSCARFIX flow records include the number of SYN, SYNACK, RST & FIN packets in the flow. In OSCARFIX we took a timeout value of 1 minute. A new flow record is generated with the first new packets crossing the DAG sniffing card. To simplify implementation, even if the flow doesn't finish in the current time interval (1 min), subsequent packets are considered as belonging to a new flow in the following minute.

Although the time measurement interval has an impact on the detection precision, an agreement at 1 minute of data collection has been adopted by all the project partners. In fact, a small interval enhances the detection delay at the cost of potentially increasing the False Alarm Rate (FAR). On the other hand, a large interval increases the detection delay. One minute was considered as a good tradeoff by the OSCAR consortium. Online implementation over Endace DAG card has been realized, and many experiments have been conducted online for accuracy analysis. Our results are encouraging in terms of accuracy and response time. All the experiments have been performed using a Ubuntu box with an Intel core 2 DUO (E4500) with 2.2 Ghz, 3 GB of RAM and 750 GB SATA disks.

In this paper, we will present the analysis results obtained over 3 traces. The first trace is made up of two hours of OSCARFIX flow records collected during online experiments in the framework of the OSCAR project. The second set of measurements is made up of an ADSL download trace ; it contains

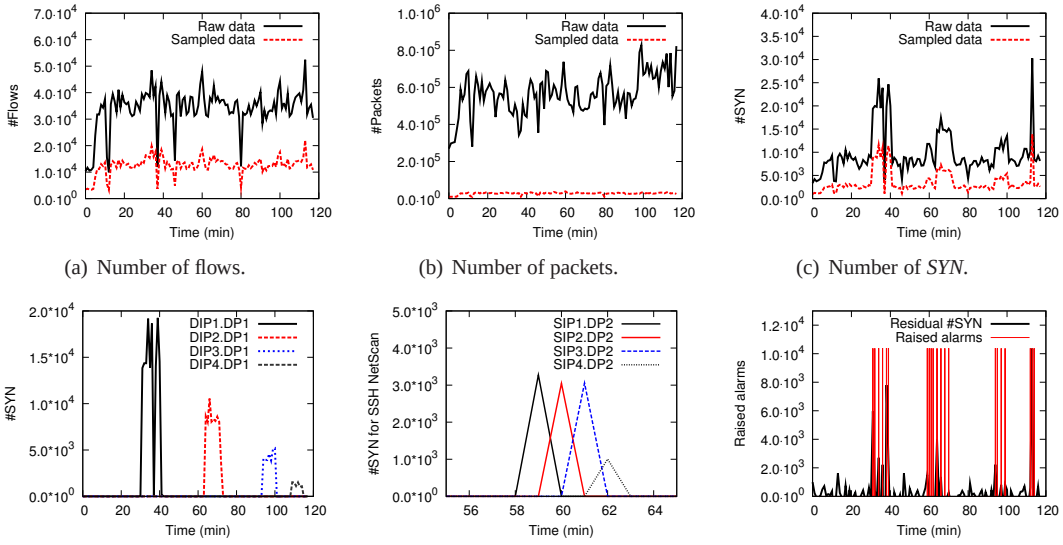
unidirectional packet traces collected during 3 hours on a geographical and technical subdivision of an ADSL access network. The OTIP trace contains 3 days of bidirectional traces collected with NetFlow on some routers of a backbone network. The main objective of parsing the last traces (ADSL & OTIP) is to check the scalability of our algorithms that is to say their ability to analyze traffic at a high data rate in real time. Each report (1 minute) must be analyzed within the next minute of data collection. The last experiment has been conducted with the aim of analyzing the performance of the algorithms ; we conducted an off-line experiment on a synthetic trace to study the influence of the various parameters on the detection and false alarm rates.

The parameters we considered for the MNP-CUSUM algorithm were the following ones: threshold value $h = 7$ in the MNP-CUSUM algorithms, weighting factor $\alpha = 0.9$ in the EWMA algorithms (estimation of the sample mean and variance), standard deviation scaling factor $c = 0.5$ in the update procedure of the MNP-CUSUM algorithms. Sketches parameters for MLRS were $P = 12$ ($l = 4$) for keys with 48bits, and $P = 14$ ($l = 5$) for keys with 64bits. CMS parameters were: $w^{'} = 4096$ and the number of hash functions $d = 4$ chosen from the set of 2-universal hash functions, and with the use of tabulation [52]. The used configuration parameters of the sampling algorithm were: $B = 3$ and $k = 2$.

Experiment 1. In the first experiment, we validate and tune the model using an online experiment trace. This online experiment has been conducted using Planetlab machines, and with the collaboration of many French research laboratories (project partners). The data trace corresponds to traffic collected at NOC in Figure 4, and each minute is the concatenation of OSCARFIX reports collected, using DAG cards with GPS-synchronized timestamp, over the experimental network of each partner. These reports are exported to NOC over UDP sockets. Reports contain much background traffic (HTTP, SSH, etc.) with mainly P2P traffic. During this experiment, well known attacks type/instant have been generated by FT to test the detection efficiency of the proposed algorithms. Figures 5(a), 5(b) & 5(c) display the variation of the total number of: flows, packets, and SYN before and after sampling. We can obviously conclude from Figure 5(c) the efficiency of the sampling algorithm in reducing legitimate variations and the amount of traffic to process. After the application of our proposed approach over this trace, we identify 4 victims of SYN flooding attacks, and 4 hosts scanning the network for an SSH server. The number of malicious SYN received by the four different victim servers are shown in Figure 5(d), where we can observe 4 attacks of different intensity. The number of SYN received by victims (10.0.0.1-4:18019) are obtained by filtering the trace to extract flow records with the identified $DIP.DP$ using our approach. We demonstrate the ability to detect low intensity attacks. In this trace, we also identify NetScan against SSH server. The number of SYN sent by scanners are displayed in Figure 5(e), where even for a small intensity, the attack has been detected and identified. The total number of SYN as well as raised alarms are shown in Figure 5(f). It is important to note that we had 0 false positives and 0 false negatives reflecting the efficiency of our proposed approach. The response time for the analysis of the 2 hour trace is less than one minute.

As we are collecting unsampled flow-records, we evaluate the impact of the used sampling technique on the precision of the anomaly detection algorithm. To compare the results, we count the number of raised alarms before and after sampling. We have one additional raised alarm after sampling during the last flooding attack. This is due to the ability of the sampling technique used to magnify suspect flows.

Furthermore, to obtain a comparison, we apply single channel non parametric CUSUM over the raw sequence which results from the aggregation of all flows into one time series. The result is very interesting and deserves to be noted. First, all NetScan attacks were not detected due to aggregation, because low intensity traffic fluctuations are not observable. Second, we had 3 false alarms even when tuning the parameters h and c. Either the last 2 flooding attacks were missed after tuning, or we had a larger number of false alarms. On the other hand, only one false alarm was obtained by applying single



(a) Number of flows. (b) Number of packets. (c) Number of *SYN*.

(d) Malicious *SYN* for DDoS. (e) Malicious *SYN* for NetScan. (f) Number of SYN & raised alarms.

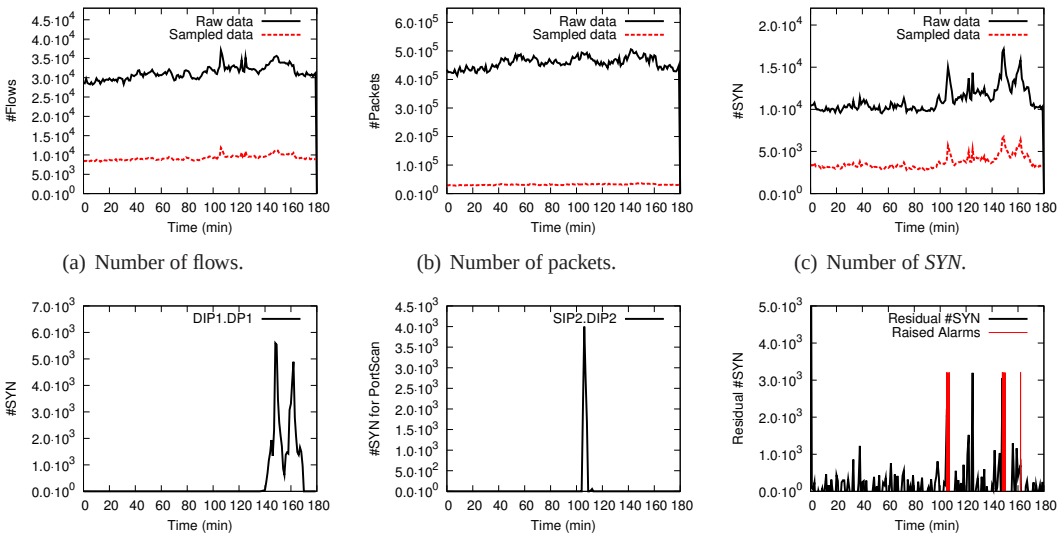
Figure 5. Analysis results for the online experiment.

channel CUSUM over the sampled traffic, where many legitimate deviations have been smoothed or discarded by the proposed sampling technique.

Experiment 2. Since the performance observed over the previous 2 hours trace can not be generalized without further analysis, we consider 3 hours of unidirectional packet level capture (pcap) with anonymized IP addresses over an ADSL infrastructure, and we transform this trace into *OSCARFIX* flow records. We carry out the same analysis and manual verification as in the first experiment. In fact, we conduct the same analysis over upload and download ADSL traces, but due to space limitations, we omit to comment on the results over the download trace since they are very similar to those over the upload trace.

Figures 6(a), 6(b) & 6(c) present the total number of flows, packets, and SYN before and after sampling. One SYN flooding attack and one PortScan have been identified by our approach. Figure 6(d) shows the number of SYN received by the identified victim server of distributed SYN flooding ($DIP1.DP1 = 97.65.192.238 : 35415$), and figure 6(e) displays the number of SYN generated for PortScan by $SIP2 = 240.178.148.21$ to scan the ports of $DIP2 = 97.68.23.88$. After filtering, we get only one additional alarm for SYN flooding at its end. As the only available information about this trace is P2P contents with some attacks, this trace may contain other undetected anomalies and we can not conclude to 0 false negatives, but we can conclude to 0 false positive. The identified SYN flooding additional alarm has been manually verified during our analysis, and also detected by other partners. The PortScan is verified manually by flow-records extraction and verification. Figure 6(f) shows the time series of the total number of SYN for sampled flows, and the raised alarms. The response time for the analysis of the 3 hours ADSL trace is less than 5 minutes.

Experiment 3. This trace is exported from CISCO routers (with Netflow v5) to a central collector (NOC), whose role is to store the received records in a database/file after anonymizing IP addresses. It contains 3 days of flow records ($\sim 896.10^5$ flows) and has a size of $\sim 6.9GB$.



(a) Number of flows. (b) Number of packets. (c) Number of *SYN*.

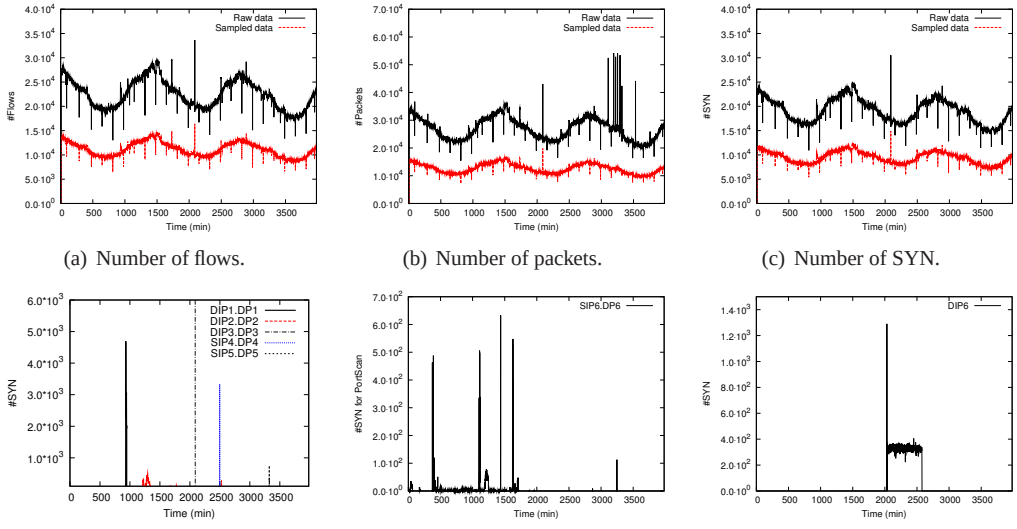
(d) Malicious *SYN* for DDoS. (e) Malicious *SYN* for PortScan. (f) Number of SYN & raised alarms.

Figure 6. Analysis results for ADSL download trace.

Figures 7(a), 7(b) and 7(c) show the variation of the total number of flows, packets and SYN before and after sampling during the 3 days. After the application of our proposed framework over traces to uncover attacks, we compare our results with other partners' results. We conclude to one misdetection after comparison and manual verification. Afterward, we isolate the number of connection requests received by each identified victim at the specific port as shown in figure 7(d) for manual verification. The number of SYN received by victim (attack missed by our approach) is presented in figure 7(f). The manual verification of missed attack shows a TCP flooding at different ports, and this explains its misdetection. Furthermore, the detection of flooding at different ports can be achieved by monitoring the DIP instead of $DIP.DP$. Also, we notice the presence of one NetScan by $SIP = 224.87.77.70$ with $DP = 65506$ (figure 7(e)). Manual verification of *OTIP* trace proves the NetScan by the given SIP at raised alarm instants. It is worth noting that response time for analyzing the whole 3 days *OTIP* trace is about a few minutes for the 3 days trace over a Pentium 2.2 Ghz with 3 GB of RAM memory.

The previous plots over real IP traces demonstrate that the number of anomalous source/destination adress pairs (6 abnormal behaviors) is so small with respect to the whole number of collected records in the trace, that providing only an attack instant without further information is not enough to uncover culprit flows. Our proposed approach identifies the bad flows and helps understanding the malicious activities behind these flows.

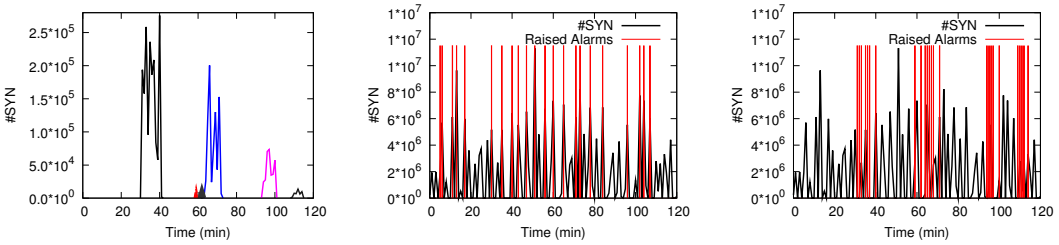
Experiment 4. This experiment is conducted in order to study the accuracy of the proposed framework and to test its sensitivity with respect to the parameters of the detection algorithm. We begin our evaluation by a comparison between CUSUM over raw data and CUSUM over sketch. By raw data we mean that the input of the CUSUM algorithm is the time series of aggregated number of SYN. We use an IP trace of 2 hours, with 6 known anomalies instant & type (4 DDoS and 2 NetScan attacks) as shown in figure 8(a). The 2 scan attacks are represented by filled curves in figure 8(a). The alarms raised by single channel NP-CUSUM over raw data are shown in figure 8(b), where we can



(a) Number of flows. (b) Number of packets. (c) Number of SYN.

(d) Malicious *SYN*. (e) Malicious *SYN* for NetScan. (f) Missed attack.

Figure 7. Analysis results for the OTIP trace.

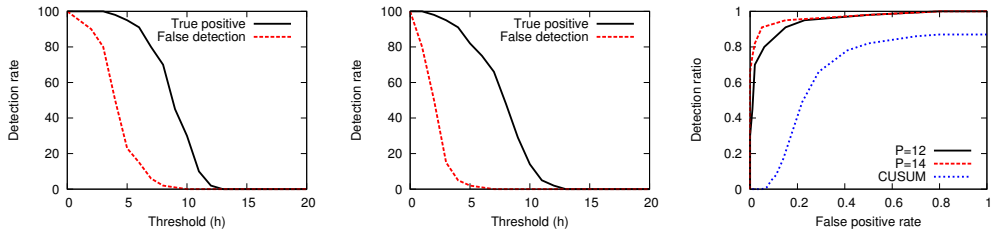


(a) DDoS & NetScan attacks. (b) CUSUM raised alarms. (c) MNP-CUSUM over sketch raised alarms.

Figure 8. Comparison between CUSUM over raw data & CUSUM over sketch.

observe false alarms due to variations in the aggregated number of SYN, and miss detection of low intensity attacks, which evade the detector after aggregation of the whole traffic into one time series. The 2 NetScan attacks are not detected when aggregating the whole traffic in one time-series, because the change is smoothed by the aggregation of the number of flows. Furthermore, to detect DDoS with small intensity attacks, a low threshold value for CUSUM is required. However, low threshold value in CUSUM algorithm incurs high false alarm rate with the detection of only 4 over the 6 existing attacks. Figure 8(c) shows the alarms raised by the MNP-CUSUM over sketch. A finer grained analysis results from the using many channels in order to monitor the traffic. This clearly reduces the number of false alarms, and increases the hit ratio of low intensity attacks.

The efficiency of an anomaly detection algorithm is usually described by two values: the power of the test (or detection rate) and the false alarm rate (FAR). In our context false alarms are legitimate flows that are classified as malicious, whereas the power of the test is the proportion of malicious flows that



(a) $P = 12$. (b) $P = 14$. (c) $P_{TP} = f(P_{FA})$.

Figure 9. $P_{TP} = f(h)$, $P_{FA} = f(h)$ and ROC for *CUSUM* & for *MLRS* with $P = 12$ and $P = 14$.

are effectively detected as malicious. There is an inherent tradeoff between power and FAR. Indeed, in an attempt to detect malicious flows with a higher probability one could be tempted to decrease the detection threshold but this would automatically result in a higher FAR. The ROC (Receiver Operating Characteristics) curve is usually used to depict such a tradeoff between power and FAR.

In our analysis we conduct off-line and many times the same experiments with different values of detection threshold h in order to test the impact of the threshold h over the power of the test and over the FAR. We also test the impact of the *MLRS* sketch width P, considering different values of P. Because of the lack of public and well documented traces with well known attacks, we use the *OTIP* trace (the largest trace with $6.9GB$ of data) as background traffic in which we manually delete the previous anomalies, and instead insert 100 SYN flooding attacks of different intensities in different times. Firstly, we apply CUSUM algorithm over this trace by aggregating whole flows in one time series. Afterward, we apply our implementation over the same trace while changing the value of the parameters h and P.

Power and FAR values are easily established because we know in advance the IP address of the victim servers, the number of existing attacks, and their instants. The power or true positive ratio P_{TP} is the number of detected attacks divided by the total number of existing ones (100). The false alarm rate P_{FA} is the percentage of raised alarms that did not correspond to real attacks. Figure 9 shows the variations of the power and the FAR as functions of the detection threshold, $P_{TP} = f(h)$ (figure 9(a)), and $P_{FA} = f(h)$ (figure 9(b)) as well as the ROC curve (figure 9(c)) for two values of the *MLRS* width ($P = 12$ and $P = 14$). As it was expected the FAR as well as the power decrease as the threshold increases. Hence, a tradeoff between false alarm rate and detection rate must be found to control the sensitivity of the test and prevent false alarms. From this study it seems that a good choice of operating point is to select $h \simeq 7$ for $P = 12$ (and $h \simeq 5$ for $P = 14$) since for these values the detection rate is high while keeping a low FAR. We also notice that a large sketch width value upgrades the performance as displayed by ROC curves on figure 9(c). Figure 9(c) shows also the ROC curve resulted from the application of single channel CUSUM over the same trace while changing the parameter h. We notice that low intensity attacks are not detected after aggregation of whole traffic in one time series, and false alarms continue to raise even with large threshold value, that was unable to detect any existing attacks. Thus is due to the high variation in the aggregated traffic pattern.

6. CONCLUSION

In this paper, we propose a new framework that integrates multi-stage sketch and multi-chart CUSUM for anomaly detection over high speed links. The proposed framework is able to automatically pinpoint the IP flows responsible for anomaly, through exploiting the matrix index in an additional multi-layer reversible sketch. The proposed approach consists of three stages: data reduction, anomaly detection and classification. The contributions are: data reduction when collecting flow records for bandwidth saving and analysis complexity reduction, software efficient sketch inversion method, making up overall an efficient algorithm to uncover hidden anomalies in the overall traffic.

It is obvious that worm signatures are unknown in their outbreak phase, and as some polymorphic worms (change its signatures to evade detection) use encryption with different keys and different encryption algorithms for every instance, it becomes a challenge for a signature based IDS to detect them. However, worms spreading phase tend to have a large number of destinations (NetScan) to infect all vulnerable systems, and thus can be identified by our proposed approach.

We proved that our approach is effective through implementation and testing on real traces with DDoS & Scan attacks. The sampling technique used discards many deviations generated by legitimate flows, and thus reduces the false alarm rate. False positive reduction is the most important factor in measuring the performance of any detection system. The use of sketch and MNP-CUSUM increases the accuracy and reduces the detection delay with respect to the aggregation of all the traffic in one time series. We have shown the ability to detect hidden anomaly in overall traffic, and to reduce the false positives. Furthermore, online experiments have proven the effectiveness of the proposed approach as well as the early detection of attacks.

The proposed method is easily decentralized due to the linear property of sketch values. Ongoing work will concern the decentralized version of the proposed approach, and the reduction of the size of exchanged sketch information between different monitoring nodes in different layers. Decentralized anomaly detection is indeed a very important issue since many attacks are deployed over the Internet and it is necessary for the different monitoring points to manage to exchange information. These monitoring points can be located for example in peering points between Autonomous Systems (AS) or in different AS. The exchange of information between AS raises new challenges concerning for example end-user privacy related questions, or trustiness between the different organizations.

Acknowledgment

This work has been partially funded by the French National Research Agency through the OSCAR project. The authors would like to thank the OSCAR consortium for their participation to some of the experiments related in this paper. They would also like to thank the anonymous reviewers for some useful feedback as well as the help of Fabio Ricciato who shepherded this article.

REFERENCES

1. Count-Min sketch source code. http://www.cs.rutgers.edu/∼muthu/massdal-code-index.html.
2. National Laboratory for Applied Network Research. http://pma.nlanr.net/.
3. Optimized RC4 code. http://www.zengl.net/freeswan/.
4. E. Ahmed, A. Clark, and G. Mohay. A Novel Sliding Window Based Change Detection Algorithm for Asymmetric Traffic. In *Proceedings of the IFIP International Conference on Network and Parallel Computing (NPC'08)*, pages 168–175, 2008.

5. E. Ahmed, A. Clark, and G. Mohay. Effective Change Detection in Large Repositories of Unsolicited Traffic. In *Proceedings of the 4th International Conference on Internet Monitoring and Protection (ICIMP'09)*, pages 1–6, 2009.
6. G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou. Network Anomaly Detection and Classification via Opportunistic Sampling. *IEEE Network*, 23(1):6–12, 2009.
7. G. Androulidakis and S. Papavassiliou. Improving Network Anomaly Detection via Selective Flow-Based Sampling. *IET Communications Journal*, 2(3):399–409, 2008.
8. P. Barlet-Ros, G. Iannaccone, J. Sanjuàs-Cuxart, D. Amores-López, and J. Solé-Pareta. Load Shedding in Network Monitoring Applications. In *Proceedings of the USENIX Annual Technical Conference (ATC'07)*, pages 1–14, 2007.
9. C. Bo, B.-X. Fang, and X.-C. Yun. A new approach for early detection of Internet worms based on connection degree. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, pages 2424–2430, Guangzhou, China, 2005.
10. D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina. Impact of packet sampling on anomaly detection metrics. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC'06)*, pages 159–164, 2006.
11. J. D. Brutlag. Aberrant Behavior Detection in Time Series for Network Monitoring. In *Proceedings of the 14th USENIX conference on System administration (LISA '00)*, pages 139–146, 2000.
12. S. Bu, R. Wang, and H. Zhou. Anomaly Network Traffic Detection Based on Auto-Adapted Parameters Method. In *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, pages 601–607, 2008.
13. T. Bu, J. Cao, A. Chen, and P. Lee. A Fast and Compact Method for Unveiling Significant Patterns in High Speed Networks. In *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 1893–1901, May 2007.
14. T. Bu, A. Chen, S. VanDerWiel, and T. Woo. Design and Evaluation of a Fast and Robust Worm Detection Algorithm. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM'06)*, pages 1–12, 2006.
15. C. Callegari, S. Giordano, M. Pagano, and T. Pepe. On the Use of Sketches and Wavelet Analysis for Network Anomaly Detection. In *Proceedings of the 1st International Workshop on Traffic Analysis and Classification (TRAC)*, 2010.
16. M. Charikar, K. Chen, and M. Farach-Colton. Finding Frequent Items in Data Streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP '02)*, pages 693–703, 2002.
17. B. Claise, S. Bryant, G. Sadasivan, S. Leinen, and T. Dietz. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101, Jan. 2008.
18. B. Claise, G. Sadasivan, V. Valluri, and M. Djernaes. Cisco Systems NetFlow Services Export Version 9. RFC 3954, Oct. 2004.
19. G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Diamond in the Rough: Finding Hierarchical Heavy Hitters in Multi-Dimensional Data. In *Proceedings of the 23rd ACM SIGMOD*, pages 155–166, 2004.
20. G. Cormode and S. Muthukrishnan. What's New: Finding Significant Differences in Network Data Streams. In *Proceedings of IEEE Infocom*, pages 1534–1545, 2004.
21. G. Cormode and S. Muthukrishnan. An Improved Data Stream Summary: the Count-Min Sketch and its Applications. *Journal of Algorithms*, 55(1):58–75, April 2005.
22. G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho. Extracting Hidden Anomalies using Sketch and Non Gaussian Multiresolution Statistical Detection Procedures. In *Proceedings of the ACM SIGCOMM Workshop on Large-Scale Attack Defense (LSAD'07)*, 2007.
23. Z. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemoto, and N. Kato. Combating Against Attacks on Encrypted Protocols. In *Proceedings of the IEEE International Conference on Communications, 2007 (ICC'07)*, pages 1211–1216, 2007.
24. W. Feng, Z. Zhang, Z. Jia, and Z. Fu. Reversible Sketch Based on the XOR-Based Hashing. In *Proceedings of the Asia-Pacific Conference on Services Computing (APSCC '06)*, pages 93–98, 2006.
25. S. Fluhrer and D. McGrew. Statistical Analysis of the Alleged RC4 Keystream Generator. In *Proceedings of the 7th International Workshop on Fast Software Encryption (FSE '00)*, pages 19–30, 2001.
26. L. He, B. Tang, and S. Yu. Available bandwidth estimation and its application in detection of DDoS attacks. In *Proceedings of the 11th IEEE Singapore International Conference on Communication Systems (ICCS'08)*, pages 1187–1191, 2008.
27. N. Hohn and D. Veitch. Inverting Sampled Traffic. *IEEE/ACM Transactions on Networking*, 14(1):68–80, 2006.
28. L. Huang, X. Nguyen, M. Garofalakis, and J. M. Hellerstein. Communication-Efficient Online Detection of Network-Wide Anomalies. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 134–142, 2007.
29. J. Kang and J.-Y. Zhang. Application Entropy Theory to Detect New Peer-to-Peer Botnet with Multi-chart CUSUM. In *Proceedings of the 2nd International Symposium on Electronic Commerce and Security (ISECS'09)*, pages 470–474, 2009.
30. R. Kawahara, T. Mori, N. Kamiyama, S. Harada, and S. Asano. A Study on Detecting Network Anomalies Using Sampled Flow Statistics. In *Proceedings of the International Symposium on Applications and the Internet Workshops*, 2007.
31. H. Kim, B. Rozovskii, and A. Tartakovsky. A Nonparametric Multichart CUSUM Test for Rapid Intrusion Detection. *International Journal of Computing and Information Science*, 2(3):149–158, 2004.
32. B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based Change Detection: Methods, Evaluation, and Applications. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC'03)*, pages 234–247, 2003.
33. A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '04)*,

pages 219–230, 2004.
34. A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies Using Traffic Feature Distributions. *SIGCOMM Comput. Commun. Rev.*, 35(4):217–228, 2005.
35. P. Lee, T. Bu, and T. Woo. On the Detection of Signaling DoS Attacks on 3G Wireless Networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pages 1289–1297, 2007.
36. C. Levy-Leduc and F. Roueff. Detection and localization of change points in high-dimensional network traffic data. *Annals of Applied Statistics*, 2009.
37. X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and Identification of Network Anomalies Using Sketch Subspaces. In *Proceedings of the 6th ACM SIGCOMM on Internet measurement (IMC '06)*, pages 147–152, 2006.
38. B. Lim and M. Uddin. Statistical-Based SYN-Flooding Detection Using Programmable Network Processor. In *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, pages 465–470, 2005.
39. W. Lu and A. Ghorbani. Network Anomaly Detection Based on Wavelet Analysis. *EURASIP Journal on Advances in Signal Processing*, pages 1–16, 2009.
40. J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is Sampled Data Sufficient for Anomaly Detection? In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC'06)*, pages 165–176, 2006.
41. D. Moore, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. In *Proceedings of the 10th conference on USENIX Security Symposium (SSYM'01)*, pages 9–22, 2001.
42. G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement (IMC '08)*, pages 151–156, 2008.
43. V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. In *Computer Networks*, volume 31 (23–24), pages 2435–2463, 1999.
44. H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for Traffic Anomaly Detection. In *Proceedings of the ACM SIGMETRICS'07 Conference*, pages 109–120, 2007.
45. M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In *LISA '99: Proceedings of the 13th USENIX conference on System administration*, pages 229–238, Berkeley, CA, USA, 1999.
46. O. Salem, S. Vaton, and A. Gravey. An Efficient Online Anomalies Detection Mechanism for High-Speed Networks. In *IEEE Workshop on Monitoring, Attack Detection and Migitation (MonAM 2007)*, November 2007.
47. R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, E. Parsons, Y. Zhang, P. Dinda, M.-Y. Kao, and G. Memik. Reverse Hashing for High-speed Network Monitoring: algorithms, evaluation, and applications. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM 06)*, pages 1–12, April 2006.
48. R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, Y. Zhang, P. Dinda, M.-Y. Kao, and G. Memik. Reversible Sketches: Enabling Monitoring and Analysis Over High-Speed Data Streams. *IEEE/ACM Transactions on Networking*, 15(5):1059–1072, Oct. 2007.
49. V. A. Siris and F. Papagalou. Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04)*, volume 4, pages 2050–2054, Dallas, USA, 2004.
50. A. Tartakovsky and K. Hongjoong. Performance of Certain Decentralized Distributed Change Detection Procedures. In *Proceedings of the 9th International Conference on Information Fusion*, pages 1–8, 2006.
51. A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. Detection of Intrusion in Information Systems by Sequential Change-Point Methods. *Statistical Methodology*, 3(3):252–340, 2006.
52. M. Thorup and Y. Zhang. Tabulation Based 4-Universal Hashing with Applications to Second Moment Estimation. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, New Orleans, Louisiana, USA, January 2004.
53. H. Wang, D. Zhang, and K. G. Shin. Detecting SYN Flooding Attacks. In *Proc. of IEEE Infocom'02*, 2002.
54. H. Wang, D. Zhang, and K. G. Shin. SYN-dog: Sniffing SYN Flooding Sources. In *Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS'02)*, pages 421–429, 2002.
55. H. Wang, D. Zhang, and K. G. Shin. Change-Point Monitoring for the Detection of DoS Attacks. *IEEE Trans. On Dependable and Secure Computing*, 1(4):1993–2004, 2004.
56. G. Yan, L. Cuellar, S. Eidenbenz, and N. Hengartner. Blue-Watchdog: Detecting Bluetooth Worm Propagation in Public Areas. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'09)*, pages 317–326, 2009.
57. N. Ye, S. Vilbert, and Q. Chen. Computer intrusion detection through EWMA for autocorrelated and uncorrelated data. *IEEE Transactions on Reliability*, 51(1):75– 82, March 2003.
58. J. Yu, H. Lee, M.-S. Kim, and D. Park. Traffic flooding attack detection with SNMP MIB using SVM. *Computer Communications*, 31(17):4212–4219, 2008.
59. Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund. Online Identification of Hierarchical Heavy Hitters: Algorithms, Evaluation, and Applications. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 101–114, 2004.

Blockmon: A High-Performance Composable Network Traffic Measurement System

Andrea di Pietro[⋆] Felipe Huici[§] Nicola Bonelli[⋆] Brian Trammell[‡]
Petr Kastovsky[*] Tristan Groleat[†] Sandrine Vaton[†] Maurizio Dusi[§]

[⋆]University of Pisa, [§]NEC Europe Ltd., [‡]ETH Zurich
[†]Institut Telecom, [*]INVEA-TECH

Abstract

As the growth of Internet traffic volume and diversity continues, passive monitoring and data analysis, crucial to the correct operation of networks and the systems that rely on them, has become an increasingly difficult task. In this paper we present Blockmon, a novel composable measurement system designed to enable measurement of ever larger and more diverse networks. Unlike existing approaches, Blockmon combines the flexibility that allows for a wide range of traffic monitoring and data analysis with the necessary mechanisms to yield high performance on today's modern multi-core hardware.

We present the design of the system, focusing on the mechanisms that enable this flexibility and performance. We present experimental results demonstrating Blockmon's performance, running simple analyses at 10Gb/s line rate without loss on commodity hardware; and compare its performance with that of existing programmable measurement systems, showing significant improvement (as much as twice as fast) especially for small packet sizes. We further demonstrate Blockmon's applicability to measurement and data analysis by implementing and evaluating three sample applications: a flow meter, a TCP SYN flood detector, and a VoIP anomaly-detection system. Blockmon is available under the BSD license.

This work was partly funded by the EU FP7 DEMONS (257315) project.

1 Introduction

Two salient trends have dominated Internet-scale monitoring and measurement over the past decade: the continued growth of the Internet, both in terms of attached nodes and total data transferred; and the diversification of devices attached to and applications running on the network. These trends are evidence of the Internet's success as a platform for communication and innovation, but they continue to increase the difficulty of the monitoring, measurement, and traffic analysis activities crucial to ensure security, quality of service, and future operational planning.

Growth in the volume of global IP traffic on the Internet continues unabated, doubling every two to three years. This traffic is also growing increasingly diverse, with mobile data traffic doubling annually; these trends are forecast to continue through 2015 [4]. This puts a heavy burden on monitoring systems, since they have to be flexible enough to adapt to this traffic diversity, and powerful enough to be able to capture and analyze large amounts of traffic in "measurement real-time"; that is, one second of traffic must on average take less than one second of total processing.

These challenges point to the need for a high-performance, yet easily-extensible solution. In this paper we present Blockmon, a system for supporting high-performance *composable measurement*: building network measurement applications out of small, discrete blocks.

1.1 Related Work

Systems work in network measurement is by no means a new field; as long as there has been an Internet, engineers and researchers have tried to measure it. Blockmon takes inspiration from three areas of work here: modular networking, programmable measurement, and high-speed packet capture.

The modular principles in Blockmon were inspired by the Click modular router [15], which has a similar design but can only perform packet-based processing, and so is only optimized for routing; we will show the effects of such optimization in a later section in a head-to-head evaluation with Blockmon.

While lots of measurement tools exist, few are flexible enough to adapt to new application areas or changing traffic. One such programmable tool is CoMo [11], which introduced the concept of a monitoring plugin by which a monitoring application is written as a set of callback functions to be called by the framework. Blockmon generalizes this concept to a message-passing architecture, described in section 2. ProgME [22] specifies a runtime-programmable network flow aggregator, configured using a declarative language based upon set algebra. RTC-Mon [9] provides a framework for building monitoring applications like CoMo, but its architecture is similarly limited to the use cases for which it was designed. In section 4 we present results to show how Blockmon compares to some of these.

Performance work in network measurement has largely focused on getting packets off the wire as fast as possible, as in [2] and PF_RING [8]; this latter even contains a basic, if inflexible, programmable measurement system. For 10Gb/s and faster links, packet capture is often enhanced through hardware acceleration as on Endace DAG or Napatech cards. NetworkDVR [3] follows another approach to packet capture by deciding early in the capture process which packets to capture and which to ignore; this can also be performed by packet capture and offload cards such as INVEA-TECH's COMBO platform. More recently, advances in commodity hardware have made 10Gb/s capture in software possible: Netmap [20], PF_RING DNA [5] and PFQ [18] are examples of this.

1.2 Contributions

Blockmon contributes some key innovations to this wide base of work, focusing on applying programmable measurement to today's traffic loads and diversity while maximizing the use of modern hardware to ensure high performance. The contributions are:

- A new, flexible design for composable network measurement and data analysis, based on message-passing among blocks, associated at runtime via gates, and supporting recomposition at runtime.
- An emphasis on applying this composition to enable parallelization of measurement processing and full use of modern multi-core hardware, ensuring high performance for line-rate measurement.
- Application of a novel *wait-free rotating queue* mechanism and new C++11 features to minimize lock contention and allocation overhead, further increasing performance.
- Integration of standards-based import and export, to extend an application across multiple nodes and to allow for easy import of data from other, non-Blockmon sources.
- The release of the Blockmon system as open source software, available at http://blockmon.github.com/blockmon.

In addition, Blockmon also leverages some of the fast packet capture technologies cited in the previous section, including easy integration of specialized hardware cards; it so far supports PF_RING, PFQ [18], and COMBO hardware cards.

2 Base System

At a high-level, Blockmon provides a set of units called *blocks*, each of which performs a certain discrete processing action, for instance parsing a DNS response, or counting the number of distinct VoIP users on a link. The blocks communicate with each other by passing *messages* via *gates*; one block's output gates are connected to the

input gates of other blocks, which allows runtime indirection of messages. A set of inter-connected blocks implementing a measurement application is called a *composition*; a generic composition is shown in figure 1.

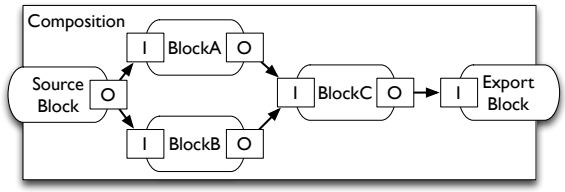


Figure 1: Example composition

```
<composition id="example">
  <block id="source" type="SourceBlock">
      <params>
      <source type="live" name="eth0"/>
    </params>
  </block>
  <block id="a" type="BlockA"/>
  <block id="b" type="BlockB"/>
  <block id="c" type="BlockC"/>
  <block id="export" type="ExportBlock">
    <params>
      <dst ip="192.0.2.20" port="4739"/>
    </params>
  </block>

  <connection src_blk="source" src_gate="pkt_out"
              dst_block="a" dst_gate="in"/>
  <connection src_blk="source" src_gate="pkt_out"
              dst_block="b" dst_gate="in"/>
  <connection src_blk="a" src_gate="out"
              dst_block="c" dst_gate="in"/>
  <connection src_blk="b" src_gate="out"
              dst_block="c" dst_gate="in"/>
  <connection src_blk="c" src_gate="out"
              dst_block="export" dst_gate="rec_in"/>
</composition>
```

Figure 2: XML corresponding to composition in figure 1

Compositions are defined using an XML format which lists the blocks and their configuration parameters, then lists the connections among their gates. The XML corresponding to the generic example composition is shown in figure 2. The Blockmon core and the blocks themselves are implemented in C++, and the system is controlled at runtime using a simple, Python-based command-line interface (CLI). In the rest of this section we describe the various parts of Blockmon in greater detail.

2.1 Blocks

As mentioned, a block performs a discrete processing action. Blocks can implement a wide range of functionality including packet capture and filtering, monitoring, anomaly detection algorithms and export capabilities. A sample of the blocks available in the base Blockmon distribution as of this writing are described in figure 3, and the list is always growing.

All blocks are derived from a common superclass. New blocks simply inherit from this class and implement at least two methods: `configure`, which receives XML representing the block's configuration parameters, and `receive_msg` which is called when a message arrives at the block. Blocks can also be invoked on periodic or one-shot timers via the `handle_timer` method, and can perform high-frequency but non-periodic asynchronous work in the `do_async` method; this last method is mainly provided for source blocks (e.g., packet capture or message import via IPFIX), which send messages but do not receive them.

2.2 Gates and Scheduling

A gate is essentially a named point on a block to allow connections between blocks at configuration time: compositions are built by defining connections between specific gates on one block and a specific gate on another. Blocks send messages via output gates, and receive messages via input gates.

There are in essence two types of input gates, which lend Blockmon its scheduling flexibility. Blockmon supports *direct* and *indirect* message passing. In the former, the sending block directly calls the receiving block's `receive_msg` method: the input gate is in this case essentially a function call. This is fast but inflexible: the receiving block runs in the sending block's thread, which will be busy with the receiving block until it finishes. This head-of-line blocking can pass all the way up the chain of directly invoked blocks, so chains of direct invocation should be avoided by using indirect message passing.

Indirect message passing is mediated by a novel wait-free, rotating queue described in section 3.2. With indi-

Block Name	Description
PcapSource	Captures packets from a local interface or pcap trace files.
PFRingSource	Captures packets from PF_RING sockets. Supports multi-queue NICs.
PFQSource	Captures packets using PFQ. Supports multi-queue NICs.
ComboSZE2Source	Captures packets from an INVEA-TECH COMBO card.
IPFIXExporter	Transcodes messages to IPFIX records, and exports them.
IPFIXCollector	Collects data via IPFIX and generates messages for appropriate records.
PacketFilter	Filters packets based on packet header fields.
PacketPrinter	Prints packets for debugging purposes.
PacketCounter	Counts received packets for debugging purposes.
IPAnon	Anonymizes the source and destination IP addresses of a packet.
FlowMeter	Assembles packets into flows with natural lifetime export.
PeriodFlowMeter	Assembles packets into flows with periodic export.

Figure 3: Sample of provided blocks.

rect message passing, each block is separately scheduled in different thread pools on different CPU cores; this allows truly parallel processing on multi-core systems without blocking or locking overhead, key to Blockmon's performance, as we will show in section 4. These queues can also buffer messages to avoid packet loss during peak load.

The two message passing models can be mixed by compositions to maximize performance. Blocks which benefit from parallelization (whether they implement parallelizable problems, or perform CPU intensive work) should generally be indirectly invoked, while thin, stateless "filter" blocks may benefit from direct invocation. Source blocks (such as packet capturers) must always be indirectly invoked via `do_async`. For implementation reasons, all input gates on a given block must be of the same type.

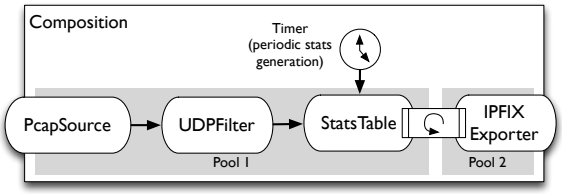


Figure 4: Example composition showing block invocation

To make things more concrete, figure 4 shows an example of how the different block and scheduling types are used in a simple composition. In this case, *PcapSource* is a source block and therefore indirectly scheduled, capturing packets from a network interface in its `do_async` method, and sending Packet messages directly to *UDPFilter*. This block filters for UDP packets, and sends results directly to *StatsTable*, which keeps statistics for received packets. All three of these blocks run in the same thread pool (see section 3.1); that is, when a packet is received, PcapSource creates a Packet message and invokes `receive_msg` on *UDPFilter*, which invokes `receive_msg` on *StatsTable* if the packet is a UDP packet.

StatsTable registers a periodic timer on configuration; Blockmon's scheduler then periodically calls `handle_timer` on *StatsTable*, sending a message containing statistics to *IPFIXExporter*. This block exports the results, and is described in more detail in section 2.5. It is indirectly scheduled, and runs in a separate thread pool from the source/counter in order to isolate network export from packet capture and counting. Periodically, the scheduler will dequeue the pending messages from the rotating queue associated with the input gate on *IPFIXExporter*, and invoke `receive_msg` for each.

2.3 Messages

The actual communication between blocks is in terms of messages. These, like blocks, are derived from a common superclass; pointers to messages are passed via the gates. The **Message** class provides a basic interface for identi-

fying message types, and for supporting import and export of messages in order to connect compositions across nodes. Messages are constant in order to ensure that they can be shared without contention among multiple blocks concurrently, and provide a tagging mechanism to allow Blocks to add small bits of data to a message in a thread-safe manner. This latter mechanism allows blocks to process high volume messages (e.g., packets) in a pipeline without incurring the overhead of having to allocate new Messages.

Messages within Blockmon dealing with timing have access to a *packet clock* derived from captured packet timestamps, to ensure that periodic processes (e.g., timeouts) run even in cases where the Blockmon system clock and the timestamps in the captured packet or flow stream have different rates (e.g., when replaying traces).

Blockmon provides a **Packet** message, which represents a packet captured from the wire by a source block and allows access to packet header and payload fields; and a **Flow** message, representing a unidirectional flow by 5-tuple and time interval. The distribution ships with additional messages designed to support processing by specific blocks (e.g., SketchMsg, which implements sketches for one of the applications presented in section 5).

2.4 Dynamic Reconfiguration

The environment in which an analysis system runs is rarely static. For example, routing changes could require traffic capture from a different interface, or information produced by an application could lead to a refined query. Blockmon supports dynamic reconfiguration: changing block parameters or the connections among running blocks, or adding or removing blocks from a composition. This allows for the implementation of control loops, where compositions can be changed based on analysis results they produce themselves. Consider a refinement of the VoIPSTREAM application presented in section 5.3, which could be configured to monitor the RTP streams of certain numbers based on changes in suspected abuse activity.

Blockmon allows changes of compositions on the fly, without losing per-block state. At reconfiguration time, messages flowing through a composition are kept in flight until the reconfiguration is complete, then processed before any new messages. Reconfiguration is available via the Python CLI.

2.5 Multi-Node Blockmon

Blockmon supports the ability to extend a single composition across a set of nodes. This allows additional scalability and flexibility of deployment. CPU-intensive processing can be split among multiple hosts, and Blockmon instances at widely separated observation points can cooperate in decentralized monitoring and correlation of events in larger networks.

This decentralization is implemented by the *IPFIXExporter* and *IPFIXCollector* blocks, respectively IPFIX [21] exporting and collecting processes. Each Message subclass which can be sent from node to node is capable of representing itself in IPFIX, and constructing new instances of itself from an IPFIX message stream, in cooperation with these blocks.

This facility also significantly increases the interoperability of our system. For example, since the Flow message can be imported from any IPFIX unidirectional flow, we can leverage data produced by existing IPFIX devices (e.g., IPFIX-enabled routers or standalone flow meters like YAF [12]) to apply Blockmon as a flow analysis tool, in addition to its native packet analysis capabilities.

The pattern followed by the *IPFIXExporter* and *IPFIXCollector* blocks can be extended to other general output formats. Current work focuses on XML marshaling, for message types which map to existing XML schemas.

3 Performance Mechanisms

Blockmon has been designed to leverage the potential of modern multi-core commodity server hardware and network interfaces. In this section we describe the optimizations used by Blockmon for high performance traffic processing. Evaluation results for these are presented in section 4.

3.1 Thread Pools and CPU Pinning

Blockmon is multi-threaded in order to take advantage of multi-core CPUs. The assignment of activities to threads and threads to CPU cores can have a large impact on performance [7]. To leverage this, Blockmon schedules work

in thread pools. Each block is assigned to a pool via the composition, and pools can be pinned to specific cores.

```
<composition id="pool_example">
  <threadpool id="pool1" n_threads="1"
              cores="2">
  <threadpool id="pool2" n_threads="3"
              cores="0-1,4">
  <threadpool id="pool3" n_threads="10">

  <block id="pkt_counter1" type="PktCounter"
         threadpool="pool1" />
  <block id="pkt_counter2" type="PktCounter"
         threadpool="pool2" />
  <block id="pkt_counter3" type="PktCounter"
         direct="true" />
  <!-- connections omitted -->
</composition>
```

Figure 5: Excerpt from a composition specifying mappings of blocks to thread pools and CPU pinnings

This model allows flexibility in terms of which block is executed on which CPU core. In the example in figure 5, `pool1` runs a single thread on its own dedicated CPU (for performance critical tasks), `pool2` shares three threads across three cores with no fixed mapping, and `pool3` provides 10 threads running on any available core. In this example, we see as well the `direct` attribute on a block, which specifies that the block is directly invoked.

This mechanism provides maximum flexibility to distribute work across the available cores in a system. The development of best practices and automated support for using this flexibility to maximize performance is the subject of future work.

3.2 Wait-Free Rotating Queues

Since indirect invocation, the key to parallelization in Blockmon, relies on passing messages from one thread to another quickly and with minimal lock contention, the design of the queues for the input gates of indirectly scheduled blocks is crucial for performance.

Here, we apply the queue design described in [18]. The gates are implemented as two queues, with one queue for writing (by the sender's thread) and one queue for reading (by the receiver's thread). The queues are rotated using atomic primitives provided by C++11 each time the Blockmon scheduler invokes an indirect block. Blockmon's wait-free rotating queues are wait-free for producers, which is crucial to keeping up with bursts of packets at line rate. Consumers only have to wait for any pending writes to complete after a swap. Experimentation shows this is a low-probability occurrence with negligible impact on performance.

This design makes synchronization fairly simple: producers reserve a slot by incrementing an atomic index, while the consumer simply swaps the active buffer (by atomically changing the index) and is granted exclusive access to the written data. Writers signal they are finished via a per-slot flag; this allows writes to complete after a buffer swap.

3.3 Batch Allocation

Since each packet captured requires a Blockmon message, dynamic memory allocation overhead [10] is a major component of the time used by compositions which read packets at line rates. Blockmon reduces this overhead by batching memory allocations of buffers used by each message into larger chunks. The batch allocations are reference-counted, such that they are automatically freed with the destruction of the last buffer in the batch. Note that this optimization is only possible with the new shared ownership constructor of the shared pointer class supported by C++11 [14], which avoids the allocation of a reference-count metadata structure for each buffer.

3.4 Efficient Message Transfer

Messages are passed using C++11 `std::shared_ptr`, so that the same message can go through different processing paths in a composition without spurious allocations or copies, with automatic reference counting. However, copying a shared pointer involves atomically decrementing and incrementing the reference counter, which can lead to high contention when a message moves from core to core. Therefore, Blockmon adopts the new C++11 object-move semantic, which allows for the transfer of shared pointers without reference count updates. This small change results in a significant performance benefit, as shown in section 4.

3.5 Pluggable Scheduling

How one schedules the various threads to run (and the blocks within them) can have a major impact on performance. Rather than trying to design an ideal scheduler for all possible compositions, Blockmon's scheduler, like that of the Linux kernel [16], is pluggable. It provides a standard scheduler that provides good performance, but allows advanced developers the ability to easily plug-in custom-built schedulers by implementing a simple interface.

3.6 Fast Capture Blocks

Fast packet capture is crucial to systems like Blockmon aimed at doing high-rate monitoring and data analysis. To this end, Blockmon provides three software-based packet capture blocks: a standard pcap-based *PcapSource* block that can capture packets from a network interface or a packet trace; a *MQPfringSniffer* block based around the PF_RING network socket [6] for higher performance; and a *PFQSniffer* block, which implements an adapter for a novel engine called PFQ [18], which better leverages multi-core architectures. Hardware-accelerated capture using INVEA-TECH Combo cards is supported by a fourth *ComboSniffer* block.

4 Evaluation

In this section we present an experimental evaluation of Blockmon in a laboratory setting, to evaluate its base performance and to explore the effects of the performance optimizations presented in section 3 in isolation, to the extent possible.

4.1 Experimental Setup

For our experiments we use a pair of servers, one running a traffic generator and the other one running Blockmon, directly connected via 10Gb wired interfaces. Each of these commodity servers costs about $2000, has a 2.66Ghz 6-core Intel Xeon X5650 with HyperThreading enabled, 12GB of DDR3 RAM, an Intel 82599EB network interface, and runs Linux kernel version 2.6.39. For packet capture we use the *PFQSource* block which yields the best performance out of the source blocks currently available in Blockmon.

A third server is used for experiments on the COMBO card; this has a 2.50GHz 4-core Intel Xeon E5420 CPU and 4GB of RAM; test traffic in this scenario comes from a hardware traffic generator. Unless otherwise stated, all experiments in this section are performed with 64-byte packets to maximize strain on the system.

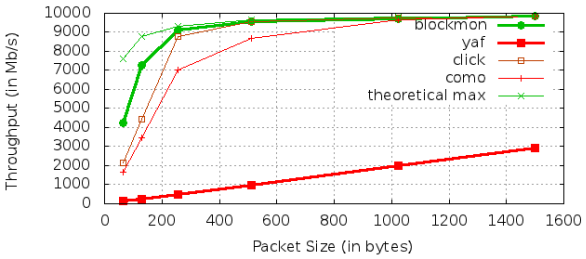


Figure 6: Performance comparison between Blockmon and other existing systems.

4.2 Comparison to Existing Measurement Systems

Before evaluating the different aspects of Blockmon's performance we wanted to measure how it stacked up against other existing programmable measurement systems. For this comparison we chose CoMo (version 1.5) since it is perhaps the most recognized system in this space; and Click (version 2.0.1), since Blockmon takes inspiration from its modular architecture. In order to compare such disparate systems, we needed to pick an application that would be simple enough to implement in each, but that would stress the systems' performance. As a result, we implemented a simple meter that keeps per-flow byte and packet counts. For further comparison, we also add YAF [12] version 2.1.2, a flow meter, to the comparison.

Although some of the measurement platforms that we used as a benchmark can make use of multiple threads, they do not support capture parallelization through explicit use of multiple hardware queues. Therefore, in order to be fair to them, we used an experimental layout where multiple kernel contexts fetch packets from multiple hardware queues and hand them over to a single capturing socket: in practice, this increases performance by using multiple kernel threads to feed packets to a single user space thread. For this, we adopted the optimal scheme described in [17], where the kernel context runs on all of the physical cores but one, which is devoted to the user–space monitoring process.

For this evaluation, we sent a synthetic stream of packets representing 1024 simultaneous UDP flows at line rate via a 10Gb/s Ethernet interface, varying the size of packets from 64 to 1500 bytes, and measured the data rate reported by each system.

As shown in figure 6, Blockmon outperforms the other systems for minimum-sized packets up to 256-byte ones, including a rate twice as fast as Click's for 64-byte packets. Blockmon also performs better than CoMo even for larger packet sizes, and roundly outperforms YAF[1] in this test. In short, Blockmon yields high performance while providing greater flexibility than existing systems.

4.3 Performance Experiments

To test the effect of the batch allocation and rotating queue optimizations, we created the Single Counter composition shown in figure 7. Each of the multiple capture blocks services one of the hardware queues on the Intel NIC, and feeds packets into a single counter block. We further assigned one CPU core to each capture block, and one for the counter. The single counter creates a bottleneck that allows us to measure the effects of the optimization.

Running this set-up produces the results in figure 8. Two logical cores are reserved for the counter block. We start our measurements with 2 cores (i.e., 2 sniffers) since setting RSS [13] on the Intel NIC to 1 causes the driver to use all of the NIC's hardware queues instead of a single one.

Applying each of the two optimizations in turn, as is the case for the "batch only" and "rotating only" curves, results in sub-optimal performance, as each optimization removes only one of the two performance bottlenecks: performance is still bound by the remaining bottleneck. Thus, the rotating queue optimization curve does not show much improvement as the number of sniffers increases and contention on the single counter becomes more severe since the memory allocation bottleneck remains. Conversely, the batch optimization curve holds steady but decreases slightly as the contention on the

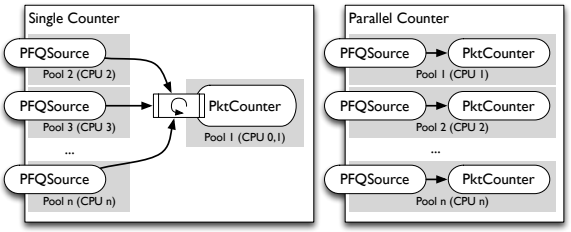


Figure 7: Compositions used in testing Blockmon performance mechanisms

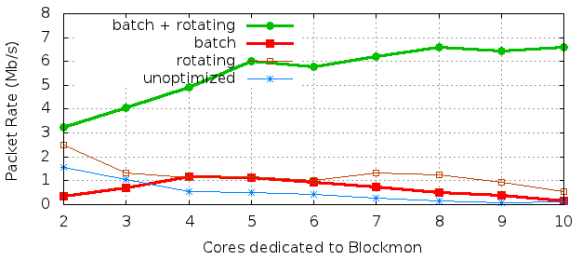


Figure 8: Effect of batch allocation and rotating queue optimizations

[1] YAF is designed somewhat differently. First, as a flow meter it must do significantly more work per packet than a simple counter. Second, its throughput is limited to about 250kpps on this traffic mix, due to its flow key hash function, which trades a high collision probability for high speed on realistic traffic mixes.

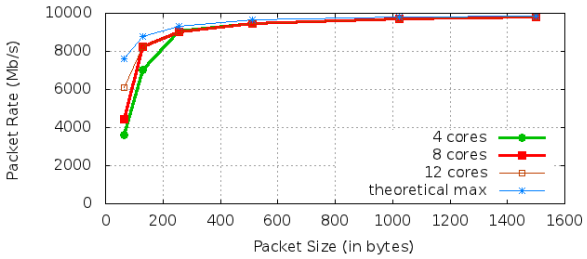


Figure 9: Performance for different packet sizes using the parallelized composition and the batching optimization.

packet counter's queue increases with the number of capture blocks. This effect is confirmed by the top curve: removing both bottlenecks provides a significant bump in performance. Note that the slight dip in performance at 6 cores is due to the fact that the last 6 cores are not physical CPUs, but rather emulated by means of the Intel Hyper-Threading technology. As a result, their contribution is lower with respect to that of actual cores.

Parallelizing packet counting, as shown in the Parallel Counter composition in figure 7, removes the single counter bottleneck, and allows us to isolate the impact of the batch allocation optimization. Batch allocation alone speeds up Blockmon by about 30% depending on the number of cores.

To show the dependence of performance on packet size, we tested the parallel composition in figure 7 with varying packet sizes. As shown in figure 9, four cores are sufficient to capture 256-byte packets at offered rate; and eight cores for 128-byte packets.

To quantify the overhead introduced by Blockmon itself, we created a simple stand-alone test application that uses the PFQ engine to capture and count packets, and compared it to Blockmon using the parallel composition in figure 7. The results show that for this simple packet application, Blockmon's flexibility has a cost of about 10% - 15%, depending on the number of cores dedicated to measurement.

We assessed the improvement brought by c++11 move semantics by writing a separate test program (due to the deep integration of move semantics in the Blockmon core, running this test on Blockmon itself would have been prohibitively difficult). The test program emulates the Message lifecycle, and compared move-based to copy-based message passing. The results of this test show up to a 20% reduction of message processing time, depending on concurrency.

5 Applications

Having shown the performance of the various mechanisms provided by Blockmon, we set out to answer the rather different question of whether Blockmon was useful for real applications. The best way to do this was by implementing three applications with Blockmon: a simple per-flow statistics meter, a TCP SYN flood detector, and a VoIP abuse detector. The first demonstrates the flexibility of composable measurement systems, including hardware-software codesign of measurement systems. The second demonstrates more complicated measurement compositions, and the third demonstrates the porting of existing complex applications to Blockmon.

5.1 Heavy Hitter Statistics

To demonstrate the ease of integrating hardware-based acceleration into Blockmon, we implemented an application for the collection of per-flow statistics for heavy hitter tracking. This application is useful on its own, and simple enough to implement twice, once in software as a simple Blockmon composition, and once in hardware with a Blockmon composition for result tabulation. This application is implemented using the two compositions shown in figure 10.

The software version uses the *FlowStats* block, which parses a flow's 5-tuple and stores statistics using a simple hash. The statistics are periodically exported to a *HeavyFlowSelector* block, which filters and exports only flows with large packet or byte counts.

The hardware-based composition mimics this, with a few differences. The sniffer is now a wrapper block called *ComboSource* which uses INVEA-TECH's COMBO card

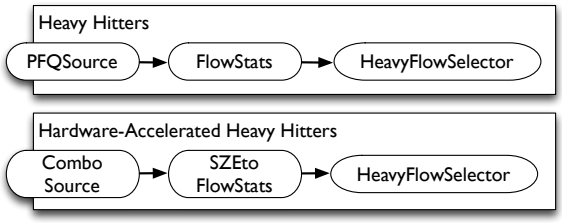


Figure 10: Software- and hardware-based compositions for the per-flow statistics application.

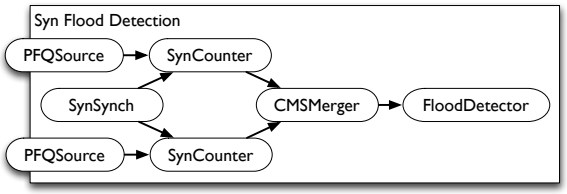


Figure 11: SYN flooding detection parallel composition (2 counters).

and keeps the per-flow statistics. This block outputs data from the hardware card in SZE2 format, where SZE2 is a zero-copy API for high-speed generic data transfers. The *SZEToFlowStats* converts the SZE2 data format into the flow statistics that *HeavyFlowSelector* expects.

We used synthetic traffic generated at 6.1 Gb/s, using minimum size packets, and had 64 of the generated flows account for 30% of the offered traffic. It is worth noting that the 30% figure is an overly pessimistic version of the 80/20 rule [19], where 20% of the flows on the Internet contribute to 80% of traffic; minimum-size packets are also a pessimistic assumption for real traffic. The not particularly well-optimized software-based composition was able to process up to 3.8Gb/s of the offered traffic.

The hardware-accelerated composition, tested with the same traffic mix at 10 Gb/s, was able to handle the offered traffic at line rate. This is not surprising, as Blockmon offloaded the flow statistics generation to the COMBO hardware in this case, and was only responsible for deframing and tabulating the results.

5.2 SYN Flooding Detection

SYN flooding is a common denial of service attack. It works by sending many TCP SYN packets to a victim, forcing it to keep potentially large amounts of state for half-opened connections.

We used Blockmon to implement an application that detects these attacks and identifies their victims. Our solution uses a Count Min Sketch (CMS) to store the number of TCP SYN packets sent to each IP address, and the multi-channel NP-CUSUM algorithm to watch all the values of the sketch and detect any abrupt changes in the number of TCP SYN packets sent to a particular IP address.

The detection is divided into different blocks brought together by the composition in figure 11. Note that for presentational reasons we only show a composition with 2 counters; this number increases in the performance evaluation with the number of CPU cores in use.

Briefly, the *SynSynch* block generates and sends a sketch during initialization to all *SynCounter* blocks, so that the size of the sketch and the hash functions used are uniform for all counters. *SynCounter* stores per-IP TCP SYN packet counts and periodically sends out a sketch with this info to *CMSMerger* which merges the sketches. Finally, *FloodDetector* detects abrupt changes (i.e., anomalies) and reports them.

We plot the performance of the TCP SYN flood detection application in figure 12. Each of the curves shows a different proportion of TCP SYN packets in the offered traffic. The results are encouraging: even at 10% (more than one million SYN packets per second) Blockmon is able to perform anomaly detection at a rate of approximately 5.5 Gb/s. To test accuracy, we manually instructed our generator to insert a burst of SYN packets at different points during the tests (recall that the detection is based on abrupt changes); each time we did so the *FloodDetector* block reported an anomaly and gave the victim's IP address.

5.3 VoIPSTREAM

To demonstrate that existing applications can be easily ported to Blockmon, and to show its flexibility in us-

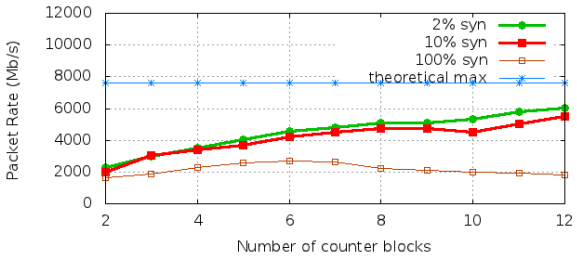


Figure 12: Performance of the SYN flood detection application.

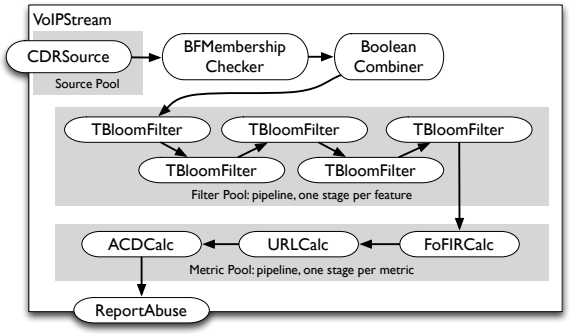


Figure 13: VoIPSTREAM Blockmon composition

ing something other than packet- and flow-based data sources, we ported VoIPSTREAM [1], a system for SIP telephony abuse detection, to Blockmon; the implementation was done by an engineer not part of the Blockmon core team. We then ran it against an anonymized Call Detail Record (CDR) data set[2] containing both VoIP- and PSTN-originated calls from a European telecom operator, and compared its performance to standalone VoIPSTREAM. The composition used for this application is illustrated in figure 13.

There are two things to note in this composition. First, the *CDRSource* block produces neither packets nor flows, and can be reused along with the associated Message subclass in other CDR-consuming applications. Second, the feature extractors (*TBloomFilter*, each with different block parameters) and metric calculators (*ACDCalc, URLCalc, FoFIRCalc*) each run in a pipeline; these pipelines offer a significant potential for parallelization.

On an eight-core 2.4GHz Xeon CPU with 24GB of RAM, standalone VoIPSTREAM processes a CDR file containing 10 million calls at a mean rate of 46.9 kCall/s over three trials. As shown in figure 14, VoIPSTREAM on Blockmon using two cores (one for decoding CDRs and one for all processing) processed the same calls at a mean rate of 41.6 kCall/s; this overhead of about 13% is on the order of that shown in section 4.3.

Of course, one key advantage of Blockmon is that it allows the efficient use of modern multi-core hardware. Simply by increasing the number of cores devoted to processing to four (one for decoding CDRs, and three for the stages in the feature pipeline, with the final metric blocks directly invoked), the rate of calls processed jumps 62% to 67.6 kCall/s, an improvement of 44% over standalone VoIPSTREAM. By using eight cores, the rate of calls improves incrementally to 70.9 kCall/s, an improvement of 51%. This demonstrates the performance advantages of porting standalone applications to Blockmon.

6 Conclusions and Future Work

We have presented Blockmon, a modular, high-performance, composable network traffic analysis system. Blockmon provides the best of both worlds: the flexibility of a modular architecture using nodes built on commodity hardware, and the high performance commonly found in more expensive, specialized hardware platforms. Blockmon can extend across multiple nodes and is dynamically reconfigurable, allowing for highly adaptive applications.

To achieve high performance we introduced a number of optimizations, and presented an extensive evaluation of their impact on performance. The results are promis-

[2]CDRs contain information such as the source and destination phone number, the time the call started, the call duration as well as the cause code or response code, which indicates whether the call was established or if an error occurred.

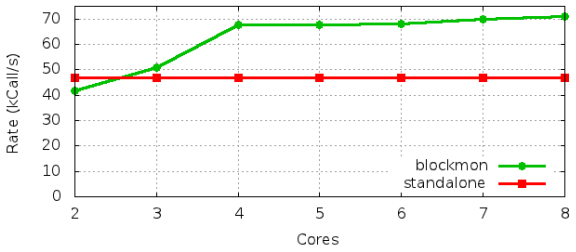


Figure 14: VoIPSTREAM performance

ing, with Blockmon able to process most packet sizes up to line rate with nontrivial applications and outperforming existing systems in the area. To illustrate possible applications of this system, we implemented a simple flow meter, a SYN flood attack detector, and a VoIP anomaly detector. The performance evaluation of each yielded rates of several gigabits per second even for minimum-sized packets.

In addition to experimenting with Blockmon installations consisting of multiple nodes, we plan to follow two specific research directions. First, the development of Blockmon has clarified the impact that optimizations in allocation and pointer handling can have with respect to performance. Future work includes the exploration of further optimizations in memory allocation specific to packet handling that attempt to make the best use of a CPU's caches.

Second, Blockmon allows very flexible scheduling with its direct and indirect block invocation as well as threadpools. However, this flexibility means that it can be difficult to decide how to choose the "right" scheduling such that a given composition yields good performance. Metrics such as queue occupancy, and CPU, memory, and cache load could be used to automatically (and adaptively depending on traffic patterns) select the best scheduling and pooling configuration.

The Blockmon distribution is available under a BSD-style license at http://blockmon.github.com/blockmon .

References

[1] BIANCHI, G., D'HEUREUSE, N., AND NICCOLINI, S. On-demand time-decaying bloom filters for telemarketer detection. *SIGCOMM Comput. Commun. Rev. 41* (September 2011), 5–12.

[2] BRAUN, L., DIDEBULIDZE, A., KAMMENHUBER, N., AND CARLE, G. Comparing and improving current packet capturing solutions based on commodity hardware. In *Proceedings of the 10th annual conference on Internet measurement* (New York, NY, USA, 2010), IMC '10, ACM, pp. 206–217.

[3] CHANG, C.-W., GERBER, A., LIN, B., SEN, S., AND SPATSCHECK, O. Network DVR: A programmable framework for application-aware trace collection. In *Proceedings of the Passive and Active Measurement Conference (PAM) 2010* (Zürich, Switzerland, mar 2010).

[4] CISCO SYSTEMS. Cisco Visual Networking Index: Forecast and Methodology. "http://www.cisco.com", June 2011.

[5] DERI, L. Direct NIC Access. `http://www.ntop.org/products/pf_ring/dna/`, December 2011.

[6] DERI, L. PF RING. "http://www.ntop.org/", May 2011.

[7] EGI, N., GREENHALGH, A., HANDLEY, M., HOERDT, M., HUICI, F., AND MATHY, L. Towards high performance virtual routers on commodity hardware. In *Proceedings of ACM CoNEXT 2008* (Madrid, Spain, December 2008).

[8] FUSCO, F., AND DERI, L. High speed network traffic analysis with commodity multi-core systems. In *Proceedings of the 10th annual conference on Internet measurement* (New York, NY, USA, 2010), IMC '10, ACM, pp. 218–224.

[9] FUSCO, F., HUICI, F., DERI, L., NICCOLINI, S., AND EWALD, T. Enabling high-speed and extensible real-time communications monitoring. In *IM'09: Proceedings of the 11th IFIP/IEEE international conference on Symposium on Integrated Network Management* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 343–350.

[10] HAN, S., JANG, K., PARK, K., AND MOON, S. Packetshader: a gpu-accelerated software router. In *Proceedings of ACM SIGCOMM 2010* (New Delhi, India, September 2010).

[11] IANNACCONE, G. Fast prototyping of network data mining applications. In *Passive and Active Measurement Conference 2006* (Adelaide, Australia, mar 2006).

[12] INACIO, C., AND TRAMMELL, B. YAF: Yet Another Flowmeter. In *Proceedings of the 24th USENIX Large Installation System Administration Conference (LISA '10)* (San Jose, California, nov 2010), pp. 107–118.

[13] INTEL. Receive side scaling on Intel Network Adapters. "http://www.intel.com/support/network/adapter/pro100/sb/cs-027574.htm", July 2011.

[14] ISO/IEC JTC1/SC22/WG21. Working Draft, Standard for Programming Language C++. `http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2011/n3242.pdf`, February 2011.

[15] KOHLER, E., MORRIS, R., CHEN, B., JAHNOTTI, J., AND KASSHOEK, M. F. The click modular router. *ACM Transaction on Computer Systems 18*, 3 (2000), 263–297.

[16] MAUERER, W. *Professional Linux Kernel Architecture*. Wrox Press Ltd., Birmingham, UK, UK, 2008.

[17] N.BONELLI, PIETRO, A. D., GIORDANO, S., AND PROCISSI, G. Packet capturing on parallel architectures. In *IEEE workshop on Measurements and Networking* (2011).

[18] N.BONELLI, PIETRO, A. D., GIORDANO, S., AND PROCISSI, G. On multi–gigabit packet capturing with multi–core commodity hardware. In *Passive and Active Measurement conference (PAM)* (2012).

[19] PSOUNIS, K., GHOSH, A., PRABHAKAR, B., AND WANG, G. Sift: A simple algorithm for tracking elephant flows, and taking advantage of power laws. In *43rd Allerton Conference on Communication, Control and Computing* (2005).

[20] RIZZO, L., AND LANDI, M. netmap: memory mapped access to network devices. In *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM* (New York, NY, USA, 2011), SIGCOMM '11, ACM, pp. 422–423.

[21] TRAMMELL, B., AND BOSCHI, E. An introduction to ip flow information export. *IEEE Communications Magazine 49*, 4 (Apr. 2011).

[22] YUAN, L., CHUAH, C.-N., AND MOHAPATRA, P. ProgME: towards programmable network measurement. *SIGCOMM Comput. Commun. Rev. 37*, 4 (2007), 97–108.

PERSPECTIVES DE RECHERCHE

Ce que l'on a fait hier n'est plus valable aujourd'hui. Ce que l'on fait aujourd'hui ne sera plus valable demain...

LES LIMITES DE LA RECHERCHE EN METROLOGIE DES RESEAUX

Pour démarrer ce chapitre je pense qu'il est utile de s'interroger sur les limites actuelles de la recherche académique en métrologie des réseaux. La communauté de recherche académique dans le domaine de la métrologie des réseaux est très active. J'en veux pour preuve les nombreux projets au niveau national et international dans le domaine de la métrologie et la littérature parfois pléthorique produite par cette communauté depuis une quinzaine d'années.

Avec le recul je pense que cette activité de recherche se traduit par trop peu de transfert technologique. Ce constat est bien sûr décevant pour la chercheuse que je suis. Mais c'est surout dommageable pour l'industrie française et européenne, pour la création d'emplois et de valeur dans ce secteur technologique. Malgré l'abondance des publications dans le domaine de la métrologie des réseaux je constate que les solutions effectivement déployées par les services opérationnels se basent peu sur les travaux académiques.

Ainsi pour estimer la matrice de trafic les opérateurs utilisent NetFlow ; ils n'optimisent de toute façon pas réellement le routage car le réseau est surdimensionné. Si on interroge les services opérationnels sur la mesure de QoS (délais, bande passante effective, etc...) on se rend compte que celle-ci n'est pas mesurée : on surdimensionne le réseau et on considère que le niveau de QoS ainsi obtenu sera suffisant. Dans le domaine de la classification de trafic la seule solution déployée se base sur de la DPI (*Deep Packet Inspection*).

Il existe un certain nombre de freins au transfert technologique dans le domaine de la métrologie des réseaux, et en particulier en ce qui concerne les méthodes se basant sur des approches statistiques. Sans prétendre à l'exhaustivité j'identifie ci-dessous un certain nombre de limitations et propose des pistes de solutions. Il serait par ailleurs intéressant d'avoir plus de retours des opérationnels sur les raisons pour lesquelles la métrologie réseaux est mature sur le plan académique mais se traduit par peu de transfert technologique.

Une première raison est que les opérateurs ne souhaitent pas communiquer sur les performances de leur réseau, que ce soit en matière de QoS ou en matière de sécurité. Il est donc difficile d'obtenir des données représentatives. Il est encore plus difficile voire même impossible de convaincre les opérateurs de mettre en place des points de capture dans leur réseau dans le cadre de projets de recherche collaboratifs. Cette opacité est une des raisons qui bloquent les avancées de la recherche dans le domaine du monitoring des réseaux de télécommunications. En réaction une association comme CAIDA [7] a pour objectif de structurer la communauté et de mettre à disposition des données publiques à analyser. Des projets comme Grenouille [13] impliquent la communauté des utilisateurs pour obtenir des statistiques sur la qualité de service offerte par les différents FAI.

Par ailleurs je pense que les services opérationnels sont méfiants vis-à-vis des approches statistiques en métrologie des réseaux. Celles-ci souffrent d'un certain nombre de préjugés : elles génèreraient trop de fausses alarmes, d'erreurs d'estimation ; elles seraient parfois complexes, difficiles à mettre en oeuvre, etc... Il est d'une part nécessaire de proposer des approches ayant des propriétés bien démontrées (taux de fausse alarme, etc...) et d'autre part de faire un effort supplémentaire de pédagogie pour expliquer ces propriétés et montrer qu'elles se traduisent par de bonnes performances en pratique. Il est également nécessaire de privilégier les approches simples et robustes, de faciliter la prise en main des méthodes en fournissant des outils (bibliothèques, etc...)

Une limitation de certaines méthodes est qu'elles nécessitent un apprentissage préalable de modèles

sur lesquelles se basent les décisions. Ces méthodes manquent parfois de stabilité et sont sensibles aux conditions sous lesquelles l'apprentissage a été réalisé. Cette limitation est à mon avis insuffisamment documentée dans la littérature. Pour pallier à ce problème il est nécessaire de développer des stratégies dans lesquelles le plan d'apprentissage est conçu de façon à s'adapter en permanence aux conditions opérationnelles (émergence de nouvelles applications, de nouvelles vulnérabilités, etc...) comme nous le proposons dans les perspectives du chapitre sur la classification de trafic. Une autre piste innovante est de développer des approches non supervisées qui découvrent certaines caractéristiques dans le trafic (clusters, anomalies, etc...) sans nécessiter d'apprentissage préalable [16].

Les travaux académiques sont souvent loin d'une solution opérationnelle ce qui est une autre limitation majeure. Par exemple la plupart des travaux ne se posent pas la question du passage à l'échelle en termes de débit des approches proposées. Il est important d'étudier les implémentations des algorithmes sous un angle pratique, en considérant les solutions logicielles ou matérielles permettant un traitement en temps réel du trafic à des débits élevés. Comme on l'a vu dans le chapitre sur la détection d'anomalies le problème des performances des implémentations logicielles et matérielles de la surveillance de trafic est au coeur des préoccupations de Blockmon [8] développé dans le cadre du projet DEMONS [11].

Enfin la mesure a le potentiel de révéler une grande richesse d'informations sur le trafic mais bien souvent on ne dispose pas d'une politique de réaction adaptée. Il me semble important de ne pas traiter la mesure indépendamment de la réaction, mais de proposer une approche intégrée. Les travaux réalisés dans le cadre du projet européen DEMONS intègrent à la fois la surveillance du trafic et les politiques de réaction. Dans DEMONS la politique de réaction est basée sur MPLS et le point de contrôle de la mitigation, interface entre le plan de détection des anomalies et le plan de réaction, se base sur le modèle ORBAC [20].

Par ailleurs dans la mesure où la mise en place d'une architecture de mesure est un facteur de coût pour un opérateur, il est nécessaire d'en démontrer les bénéfices financiers. Un scénario intéressant est celui de la QoS inter-domaine étudié dans le cadre du projet ETICS [12]. Une partie de l'Internet va évoluer vers une ressource que l'on va pouvoir louer en échange d'une garantie de débit et de QoS. Il est intéressant de prendre en compte les informations remontées par la métrologie dans ce modèle économique. Les enjeux ne sont pas uniquement des enjeux technologiques, la mesure de QoS (délai, gigue, bande passante disponible, ...) est en effet un sujet qui a été très largement étudié depuis au moins une dizaine d'années. L'enjeu est également économique et à l'heure actuelle il existe à ma connaissance très peu de travaux sur la prise en compte des informations remontées par la métrologie dans les modèles économiques. Dans le cadre de la thèse d'Isabel Amigo financée par le projet ETICS nous étudions en particulier comment la détection de violations de SLA doit être prise en compte dans la façon dont les revenus générés par des services inter-domaine avec QoS garantie sont partagés entre les différents domaines.

PLATEFORME POUR LA METROLOGIE DES RESEAUX A HAUT DEBIT

L'augmentation exponentielle des débits de trafic dans les réseaux est un défi majeur pour le monitoring. La question du passage à l'échelle des méthodes proposées est donc un problème récurrent quelque soit l'application considérée. La recherche académique dans le domaine de l'analyse de trafic est en train d'évoluer. Il y a dix à quinze ans la méthodologie adoptée par beaucoup de chercheurs consistait à obtenir des traces de trafic, issues de campagnes de mesure, puis à les analyser hors ligne afin de comprendre la structure du trafic, de développer des modèles mathématiques et des algorithmes de traitement parfois sophistiqués. L'analyse ne se faisait pas en temps réel et, dans la majorité des cas, on se préoccupait peu du temps d'exécution de ces algorithmes, de leur capacité à être exécutés sur du trafic en temps réel à un débit réaliste.

Les solutions commerciales spécialisées dans le monitoring de trafic intègrent beaucoup de mécanismes d'optimisation pour supporter la montée en débits. D'ailleurs certaines sociétés du domaine, comme la société QoSMoS [18], issue du LIP6, sont des spin-offs d'organismes d'enseignement supérieur ou de recherche publique. Il me semble qu'aujourd'hui une part croissante de la communauté académique s'intéresse aux performances des traitements en termes de débit de trafic.

Des programmes de recherche comme NetFPGA [17] initié par l'université de Stanford offrent à la communauté d'enseignement et de recherche certains outils permettant de relever ce défi. Le programme NetFPGA développe des cartes programmables orientées vers l'intégration matérielle sur FPGA d'algo-

rithmes de traitement du trafic. Il existe deux générations de cartes NetFPGA avec des interfaces à 1 Gb/sec et 10 Gb/sec même s'il s'avèrait jusqu'à récemment difficile de se procurer cette dernière version. Le programme NetFPGA bénéficiant de sponsors comme XilinX le prix des cartes les rend accessibles pour la communauté recherche et enseignement.

Depuis environ deux ans nous utilisons des cartes programmables pour intégrer différents algorithmes de traitement du trafic. Ce travail se fait en collaboration avec le département Electronique de Télécom Bretagne, et en particulier avec Matthieu Arzel. Nous utilisons les cartes NetFPGA dans le cadre de projets avec des élèves de Télécom Bretagne, de stages, et de la thèse de doctorat de Tristan Groléat. Différents types de traitement ont été intégrés sur NetFPGA ou sont en cours de développement : détection de flots massifs (Count Min Sketch, CMS), classification de flots par une méthode SVM, génération de trafic, etc ... Le travail sur la classification de trafic avec accélération matérielle a été décrit dans [14] et dans le chapitre correspondant de ce manuscrit. En plus des cartes NetFPGA à 1Gb/sec. nous avons fait l'acquisition d'une carte COMBOv2 [15]. Cette carte possède deux interfaces à 10 Gb/sec et est commercialisée par la société tchèque INVEA-TECH, spin-off de l'université de Brno et partenaire du projet européen DEMONS [11].

Nous disposons par ailleurs depuis longtemps d'une carte DAG de la société Endace [9], carte commerciale permettant la capture de 100% des paquets au débit maximum de l'interface avec un horodatage précis. Nous avons également acheté des machines puissantes pour certains traitements, par exemple un Dell Precision T7500 (2 processeurs Intel Xeon X5660), et une carte graphique NVIDIA Geforce GTX580 pour faire du calcul sur carte graphique.

Dans le cadre des études sur le monitoring de trafic j'aimerais mettre en place une plateforme à 10 Gb/sec. Différents équipements sont nécessaires : générateurs de trafic (Spirent ou IXIA), analyseurs de trafic (Agilent), routeurs CISCO, sondes NetFlow, commutateurs, Network Test Access Points (TAPs, société Comcraft), etc... Les équipements commerciaux sont onéreux quand on vise des débits de quelques dizaines de Gb/sec.

Quelques équipements commerciaux seraient très utiles pour compléter les différents modules (générateurs de trafic, sondes, etc...) que nous arrivons à développer sur NetFPGA moyennant de nombreuses heures de travail avec les étudiants de Télécom Bretagne. Un projet a été déposé dans le cadre de l'appel à projets 2011 pour Equipements d'Excellence (Equipex, [1]) mais n'a malheureusement pas été retenu. Pour le moment nous envisageons d'emprunter du matériel à nos partenaires industriels et de déposer à nouveau un projet dans le cadre d'un prochain appel.

Dans l'avenir nous devons étudier le problème plus finement en comparant les approches logicielles et les approches matérielles et en identifiant quels sont les goulots d'étranglement qui doivent bénéficier d'une accélération matérielle. Des progrès considérables ont été faits dans le domaine des architectures multi-coeurs et de la parallélisation de code. Il est important de profiter de l'accélération obtenue en répartissant certains calculs sur les coeurs de traitement pour améliorer les débits d'analyse. Dans l'avenir j'aimerais considérer d'autres solutions que les cartes FPGA pour accélérer le traitement du trafic ; les processeurs de réseau sont une autre piste intéressante que nous devrions explorer.

Nous avons pour l'instant priviligié les solutions basées sur une intégration matérielle en raison de la collaboration en cours avec le département Electronique de Télécom Bretagne. Faute de temps et de moyens humains disponibles, nous n'avons pas suffisamment exploré les approches logicielles optimisées qui permettent d'obtenir des performances très intéressantes sur des architectures non spécialisées. Le travail collaboratif réalisé dans le cadre du consortium DEMONS nous a ouvert la porte de ce domaine de recherche très intéressant. J'espère que nous pourrons par la suite bénéficier de l'expérience que nous commençons à acquérir sur le sujet pour développer cet axe de recherche à Télécom Bretagne, si des moyens suffisants sont obtenus.

L'accès à du trafic réaliste et temps réel est un point crucial pour valider nos développements. Pour cela on peut utiliser un générateur de trafic mais le trafic produit n'est pas très réaliste. L'idéal serait de pouvoir positionner un point de capture chez un opérateur de réseau. Cependant comme je l'ai expliqué plus haut il n'est pas envisageable de positionner une sonde chez un opérateur commercial. Par contre il pourrait être envisagé dans le cadre d'un projet impliquant RENATER de positionner un point de capture sur le réseau français pour la recherche et l'éducation. Ceci nous donnerait accès à des données de trafic réalistes ; en retour de quoi Renater pourrait obtenir des remontées d'information intéressantes de la part des sondes.

Faute de pouvoir accéder à un point de capture à un niveau d'agrégation plus élevé, nous avons

positionné une sonde de trafic sur le RESEL [5], réseau des élèves de Télécom Bretagne. Il permet à environ 600 élèves de se connecter à l'Internet depuis les résidences universitaires. La bande passante allouée au RESEL est de 300 Mb/sec. Cette sonde basée sur une carte NetFPGA 1G a été mise en place à l'automne 2011 après une consultation et un vote des utilisateurs du RESEL lors d'une Assemblée Générale. Nous avons bien sûr pris toutes les précautions nécessaires et nous sommes engagés dans le cadre d'un protocole d'accord à n'utiliser cette sonde qu'à des fins de recherche et à respecter la confidentialité des données des usagers du RESEL. La mise en place de ce point de capture nous donne accès à un trafic de campus réaliste en temps réel qui nous permet de mettre en place différentes expérimentations en partenariat avec les étudiants de Télécom Bretagne et en particulier les administrateurs du RESEL. Par exemple, un projet d'étudiants en cours à Télécom Bretagne , consiste à détecter en temps réel l'utilisation de P2P sur le RESEL et de mettre en place une politique de réaction adaptée.

ECONOMIE DE L'INTERCONNEXION DE RESEAUX

Actuellement les relations entre les différents acteurs de l'Internet (fournisseurs d'accès, opérateurs, fournisseurs de contenus, ...) sont extrêmement opaques. Les nouvelles applications génèrent des profits considérables pour les fournisseurs de contenus mais elles induisent une croissance très rapide des débits ce qui conduit l'Internet dans son modèle actuel au bord de l'explosion.

Les revenus générés par les nouvelles applications profitent aux fournisseurs de services mais pas directement aux opérateurs de réseaux. Les fournisseurs de services sont parfois vus, de manière un peu caricaturale, comme les "parasites" de cet écosystème. D'un autre côté c'est la diversité des services et des contenus proposés aux internautes qui rend l'Internet attractif et permet aux fournisseurs d'accès de vendre un abonnement mensuel à la majorité des foyers.

Un des enjeux de l'Internet du futur n'est donc pas un problème technique mais un problème économique. Il est nécessaire de construire un modèle économique qui ne bride pas le développement de l'Internet mais au contraire encourage tous les acteurs de cette chaîne de valeurs à participer à son développement [4].

Dans le cadre du projet intégré européen ETICS [12] [19], Isabel Amigo dont je co-encadre la thèse avec le professeur uruguayen Pablo Belzarena, étudie certains problèmes économiques relatifs à la QoS en interdomaine. Dans la perspective d'ETICS une partie de l'Internet est vu comme une alliance de domaines dont on peut composer les ressources pour acheter un service avec une garantie de bout-en-bout. Différents mécanismes de vente de ces services garantis en interdomaine peuvent être considérés. Dans le cadre de la thèse d'Isabel nous proposons une solution basée sur un mécanisme d'enchères [2]. Les enchères peuvent être résolues de manière distribuée, c'est-à-dire sans avoir recours à un tiers de confiance qui centraliserait des informations sur les capacités des domaines.

L'alliance doit périodiquement partager les ressources générées par la vente de services avec garantie de QoS. Nous concevons des mécanismes de partage des ressources qui sont équitables, incitent les acteurs à rester dans l'alliance, et à augmenter les capacités dédiées à l'alliance [3].

Par ailleurs nous allons intégrer dans nos modèles économiques des retours sur la mesure de QoS. En effet la qualité de service effectivement délivrée doit être mesurée et confrontée aux accords de service (*Service Level Agreements*, SLA). Nous allons prendre en compte la mesure des performances réalisées en envisageant une compensation monétaire reversée au client dans le cas où la QoS n'est pas réalisée. Il est nécessaire de comprendre comment le mécanisme de compensation influence la demande et donc les revenus de l'alliance de façon à optimiser ce mécanisme pour maximiser les revenus.

PROJET EUROPEEN D'ENSEIGNEMENT DES RESEAUX DANS LES LYCEES ET DANS LE SUPERIEUR

En collaboration avec différents collègues au niveau européen nous avons à moyen terme un projet qui se situe à la frontière entre les activités de recherche et d'enseignement. Il s'agit de la mise en place d'une plateforme d'enseignement de la science des réseaux de télécommunications, en utilisant des moyens technologiques et une pédagogie innovants. Le projet vise deux types de public : les scolaires et notre

publica habituel, les étudiants de niveau universitaire.

Ce projet a démarré sur deux constats. D'une part les réseaux de télécommunications sont devenus le système nerveux central de notre société. Le grand public et en particulier les scolaires ne sont pas suffisamment éduqués à l'informatique, à l'électronique, aux réseaux de télécommunications alors que ces disciplines mériteraient de devenir des matières scolaires à part entière, au même titre que la physique par exemple, du fait de leur très grand impact sociétal. L'informatique et les réseaux sont très présents dans les débats sociétaux. Le secteur des TIC représente 30% de la R&D en Europe. L'informatique a pénétré tous les domaines d'activité (médecine, transports, commerce, finances, etc...) Il est donc primordial que tous les jeunes et en particulier les futurs cadres soient formés dans le domaine et comprennent quelques principes fondamentaux sur les réseaux de télécommunications.

D'autre part l'Europe a un besoin crucial d'experts dans le domaine des réseaux. L'Europe a besoin de former des experts ayant des compétences concrètes en termes de déploiement et d'administration de réseaux, et qui par ailleurs comprennent les principes théoriques fondamentaux. Les futurs experts doivent être sensibilisés aux problématiques de l'embarqué, avoir été familiarisés à la synthèse. Sans cette formation il devient à très court terme impossible de maintenir une industrie européenne compétitive dans le secteur.

Alors que le secteur des TIC a pris une importance fondamentale en quelques décennies les enseignements dans ce domaine étaient jusqu'à très récemment absents dans les lycées d'enseignement général [10]. L'enseignement scientifique et technologique se cantonnait aux mathématiques, à la physique et aux sciences de la vie et de la terre (SVT). Le paysage commence à évoluer puisque, dans le cadre de la réforme des lycées, un enseignement de spécialité en informatique et en sciences du numérique a été introduit en terminale scientifique [6].

Une des conséquences de ces carences est une méfiance des futurs ingénieurs vis-à-vis des formations spécialisées en informatique et télécommunications. Beaucoup d'étudiants déclarent préférer une formation d'ingénieurs "généraliste" plutôt qu'une formation spécialisée en télécommunications, ce qui nuit au recrutement dans nos écoles. J'impute cette perception à une méconnaissance du secteur, liée pour partie à un enseignement insuffisant dans le secondaire. Une fois recrutés une part non négligeable de nos étudiants se détournent des TIC pour continuer à faire des mathématiques, appliquées cette fois-ci à la finance. Cet état de faits est paradoxal étant donné le gisement considérable d'emplois que représente le secteur des TIC.

Dans le secondaire un long chemin a été parcouru dans l'enseignement de la physique, avec une réflexion sur l'enseignement par l'expérimentation. Il existe toutes sortes de kits bon marché permettant de faire toucher du doigt les principes fondamentaux de la physique en se basant sur des expériences simples. Mais on ne trouve pas de support pédagogique adapté pour l'enseignement de la science des réseaux aux scolaires, probablement parce que cette science est extrêmement récente. Le chemin parcouru dans le domaine de la physique doit être parcouru dans l'enseignement des réseaux. Notre frontière à nous n'est pas l'espace mais "l'Internet du Futur" ; repousser cette frontière a le potentiel de faire rêver nos jeunes scientifiques. Un des objectifs de ce projet est donc de fournir des supports de cours adaptés à l'enseignement de la science des réseaux dans le secondaire et en particulier à l'enseignement de la science des réseaux par l'expérimentation.

D'autre part l'Europe a un besoin crucial d'experts dans le domaine des réseaux. Dans le cadre de ce projet il est envisagé d'utiliser des méthodes d'apprentissage renforcées par la technologie (*Technology Enhanced Learning*, TEL) pour stimuler la communauté universitaire par rapport à la problématique de la synthèse et pas uniquement par rapport à l'analyse. Nous souhaitons introduire de nouvelles méthodes pour enseigner les réseaux en utilisant les outils à disposition en termes de simulation, émulation, synthèse... L'objectif est de permettre aux étudiants de toucher immédiatement et de manière concrète à la réalité.

Le projet mettrait en place un nuage auquel les étudiants pourraient accéder pour émuler un réseau, par exemple en utilisant un outil comme GNS-3. L'objectif est de donner aux étudiants les capacités de développer des fonctionnalités réelles en logiciel et en matériel ; ce n'est pas une utopie, c'est ce que commencent à faire certaines universités européennes en utilisant la plaforme NetFPGA. L'objectif est également de transmettre un bagage théorique en mettant en place une communauté d'experts auprès desquels il serait possible de trouver des formations plus théoriques.

Les objectifs sont donc ambitieux : (i) maîtriser des équipements commerciaux réels (à la manière de la "Cisco Academy") (ii) comprendre la théorie : ne pas connaître uniquement les fonctionnalités mais

avoir également une méthodologie pour optimiser les paramètres, (iii) simulation et émulation au travers du Nuage avec des approches de type Software as a Service (SaaS), Platform as a Service (PaaS), etc...(iv) avoir l'opportunité de revenir à l'invention : disséquer la machine pour en construire une meilleure...

PROJET NATIONAL D'OBSERVATOIRE DE L'INTERNET

Aux USA, CAIDA (Cooperative Association for Internet Data Analysis)[7], réalise des outils et des mesures depuis 1997. Même si ce projet a une vocation mondiale, sa portée se limite essentiellement aux USA et beaucoup de mesures ne sont pas publiques. L'objectif du projet d'"Observatoire de l'Internet" est de doter la France, puis dans un deuxième temps l'Europe, d'un tel outil.

L'Internet n'est pas un objet comme les autres ; en effet, l'Internet est devenu en quelques années un élément structurant majeur pour la société. L'étude d'un objet comme Internet est donc délicate car elle doit prendre en compte à la fois des problématiques technologiques et des problématiques sociétales. L'objectif d'un observatoire de l'Internet est de superviser l'Internet et son évolution, la QoS offerte aux utilisateurs, la neutralité du réseau, l'évolution des usages, etc ...

La mise en place de cet observatoire nécessite des développements autour de différents axes que nous énumérons ci-dessous.

Points de capture La mesure doit être mise en oeuvre au niveau de différents points de capture. Il peut s'agir de mesure passive ou de mesure active. La mesure active consiste à injecter du trafic pour sonder le réseau. La mesure passive surveille le trafic de charge sans injecter de trafic de sonde. Les mesures peuvent se faire en un point ou bien nécessiter la collaboration de deux points de capture sur le réseau dans le cas de mesures de bout en bout. Pour certaines applications il est nécessaire de s'assurer d'une synchronisation précise des points de capture ; cette synchronisation peut être obtenue par GPS.

Plan de contrôle Un plan de contrôle doit être défini. Le plan de contrôle est en charge de la configuration des points de capture, du lancement des mesures, de la collection des résultats au niveau de un ou plusieurs collecteurs.

Stockage et accès aux données Des serveurs bien dimensionnés doivent être prévus pour stocker les résultats de la mesure. Les données de mesure sont le plus souvent sensibles. Elles doivent dans certains cas être anonymisées dans un souci de protection de la vie privée. Il convient par ailleurs d'élaborer des stratégies de contrôle d'accès pour protéger les données et différencier les droits d'accès des diverses catégories de participants au projet.

Analyse des données Les données doivent être visualisées. Des représentations graphiques doivent être extraites des données brutes. L'évolution temporelle et spatiale de différents indicateurs doit être étudiée. Les mesures doivent être corrélées spatialement ; les approches tomographiques permettent de déduire des informations sur l'état interne du réseau en confrontant les résultats obtenus en différents points de capture. Les événements tels que pannes ou dégradation de la QoS doivent être identifiés. Les résultats de la mesure doivent être confrontés aux modèles mathématiques de l'Internet.

Impact académique Les outils permettant de déployer la mesure et d'en analyser les résultats doivent être développés et mis à disposition de la communauté scientifique, par exemple sous la forme de logiciels libres. Les résultats des campagnes de mesure doivent également être mis à disposition de la communauté pour permettre une analyse ultérieure et une confrontation des différentes méthodes d'analyse. Les données brutes peuvent être enrichies de certains résultats d'analyse avant d'être publiées. Il peut s'agir par exemple d'une "groundtruth", une vérité "absolue" obtenue par d'autres méthodes et à laquelle les chercheurs pourront confronter les résultats de leurs propres analyses.

Impact sociétal La communauté des Internautes doit être sollicitée pour participer aux expérimentations à grande échelle. Ils peuvent participer en acceptant d'installer certains agents logiciels sur leur machine ou en remontant des retours d'expérience sur la qualité de leur accès Internet. Un effort pédagogique important doit être fait dans certains cas pour obtenir l'adhésion d'une partie des utilisateurs.

CONCLUSION

Dans ce document j'ai tenté de faire un panorama des évolutions récentes et des enjeux actuels du domaine des réseaux et de positionner la métrologie dans ce paysage. J'ai ensuite structuré ce manuscrit autour de quatre axes principaux, axes sur lesquels j'ai travaillé au cours des dernières années : la matrice de trafic, le routage robuste, la classification de trafic, et la détection d'anomalies dans le trafic. A la fin de chacun de ces chapitres j'ai présenté quelques perspectives de recherche sur ces thématiques.

Puis dans le dernier chapitre j'ai décrit quelques projets de recherche qui sont en cours de développement. Ces projets sont à des stades plus ou moins avancés. Deux d'entre eux font l'objet de thèses en cours, celles de Tristan Groléat et d'Isabel Amigo, et ont déjà donné lieu à quelques publications. Je n'ai pas dédié un chapitre particulier à ces axes sachant qu'il s'agit de travaux qui sont encore en cours et que je trouve prématuré d'en faire le bilan aujourd'hui. Dans l'annexe D de ce manuscrit j'inclue quelques publications sur des problèmes économiques liées à la QoS en inter-domaine qui sont issues du travail de thèse d'Isabel.

Deux autres projets, celui sur l'enseignement des réseaux et celui sur l'observatoire de l'Internet, sont des réflexions plus prospectives au sein d'un consortium, au niveau national pour le projet d'observatoire de l'Internet et au niveau européen pour le projet d'enseignement des réseaux. J'espère que ces projets se concrétiseront. Cela dépendra en grande partie des ressources que nous pourrons y consacrer et donc du succès que nous obtiendrons dans le cadre de futurs appels à projets sachant que je suis la seule enseignante-chercheurse permanente à travailler sur ces thématiques dans mon laboratoire.

Certains travaux plus anciens n'ont pas été inclus dans les chapitres principaux de ce document. En effet cela aurait nécessité de refaire un état de l'art complet sur ces thématiques afin de positionner ces travaux dans la littérature et de dresser quelques perspectives. Le temps a manqué pour cela, d'autant plus que certains de ces travaux correspondent à des axes de recherche que je n'ai pas maintenus.

Cependant il aurait été dommage de ne pas mentionner ces contributions, en particulier parce que certaines publications issues de ces travaux ont été très fréquemment citées dans la littérature. C'est pourquoi j'ai annexé certaines publications complémentaires à ce manuscrit.

L'annexe A présente des contributions dans le domaine des communications numériques, en particulier un article sur une méthode de détermination de la distance libre de codes correcteurs d'erreurs et un autre article sur un schéma de décodage itératif en CDMA. L'annexe B présente divers travaux sur la modélisation statistique de trafic et de canal : un article sur une modélisation de canal par chaîne de Markov cachée, un article sur l'estimation de paramètres de décalage dans un mélange de lois exponentielles, et un article sur la prédiction de trafic saisonnier par un modèle SARIMA. L'annexe C présente des travaux issus de la thèse CIFRE d'Hamza Dahmouni chez Orange Labs sur la modélisation de trafic et l'évaluation de performances dans des réseaux voix+données GPRS/EDGE. Enfin l'annexe D présente des travaux en cours dans le cadre de la thèse d'Isabel Amigo en cotutelle avec l'Institut d'Ingénierie Electrique (UdelaR, Uruguay) : un article sur la vente par un mécanisme d'enchères de services inter-domaines avec un délai de bout-en-bout garanti, et un article sur le partage entre les différents domaines qui constituent l'alliance des ressources issues de la vente de ces services.

Cette liste de travaux en métrologie des réseaux et des perspectives qui peuvent en découler est loin d'être exhaustive. Le trafic est une cible mouvante, en permanente mutation, du fait de l'émergence de nouveaux usages et du développement de nouvelles technologies. La métrologie des réseaux va devoir s'adapter à ces évolutions : l'Internet mobile, le cloud computing et d'une manière générale l'informatique distribuée (grilles, etc...), les nouvelles stratégies de distribution de contenus multimédia, en particulier dans un contexte inter-domaines. La métrologie des réseaux est donc un domaine extrêmement riche qui offre un très grand nombre de problématiques à explorer. Je me propose de continuer à contribuer au développement de ce domaine pendant quelques années et de motiver les nouvelles générations d'étudiants et de chercheurs pour ces problématiques. Merci pour votre lecture attentive...

Je n'ai que l'idée que je me fais de moi pour me soutenir sur les mers du néant...

Henry de Montherlant, Service Inutile.

Références

[1] Agence Nationale de la Recherche (ANR). Appel à Projets Equipements d'Excellence (EQUIPEX). http://www.agence-nationale-recherche.fr/investissementsdavenir/AAP-EQUIPEX-2011.html.

[2] I. Amigo, P. Belzarena, F. Larroca, and S. Vaton. Network bandwidth allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations. In *ICQT'11 : 7th International ICQT Workshop on Advanced Internet Charging and QoS Technology*, Heidelberg, Germany, 2011.

[3] I. Amigo, P. Belzarena, and S. Vaton. On The Problem of Revenue Sharing in Multi-domain Federations. In *Networking 2012*, 2012.

[4] ARCEP : Autorité de Régulation des Communications Electroniques et des Postes. *Internet est devenu un bien collectif stratégique.* Cahiers de l'ARCEP, 3ème trimestre 2010.

[5] Association Résel : réseau des élèves de Télécom Bretagne. Bienvenue sur le site du RESEL ! http://resel.fr.

[6] Bulletin Officiel spécial numéro 8 du 13 octobre 2011. Enseignement de spécialité d'informatique et sciences du numérique de la série scientifique - classe terminale. http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57572.

[7] CAIDA. CAIDA Data - Overview of Datasets, Monitors, and Reports. http://www.caida.org/data/overview/.

[8] Di Pietro, A. and Huici, F. and Bonelli, N. and Kastovsky, P. and Trammell, B. and Groleat, T. and Vaton, S. and Nuccilli, F. Blockmon : a modular system for flexible, high-performance traffic monitoring and analysis. soumis à ATC Usenix 2012.

[9] Endace. Endace DAG high-speed capture cards. http://www.endace.com/endace-dag-high-speed-packet-capture-cards.html.

[10] EPI. Association Enseignement Public et Informatique. http://www.epi.asso.fr/accueil.htm.

[11] European Commission 7th Framework Program (FP7). DEMONS : DEcentralized MONitoring for trustworthinesS . http://fp7-demons.eu/.

[12] European Commission, Information Society Technologies, Seventh Framework Program. ETICS : Economics and Technologies for InterCarrier Services. https://www.ict-etics.eu/.

[13] Grenouille.com. Grenouille : la météo du Net. http://www.grenouille.com/cest_quoi.php.

[14] T. Groleat, M. Arzel, and S. Vaton. Hardware acceleration of SVM based traffic classification on FPGA. In *IWCMC TRAC'12 : 3rd International Workshop on Traffic Analysis and Classification*, Limassol, Cyprus, 2012.

[15] INVEA-TECH. COMBO FPGA boards. http://www.invea-tech.com/products-and-services/combo-fpga-boards.

[16] J. Mazel, P. Owezarski, and Y. Labit (directeurs). *Unsupervised network anomaly detection.* Thèse de doctorat du LAAS/CNRS, LAAS/CNRS, Toulouse, 2011.

[17] NetFPGA. NetFPGA : a line-rate, flexible, and open platform for research, and classroom experimentation. http://netfpga.org/.

[18] QoSMoS. Qosmos : your network is information. http://www.qosmos.com/french.

[19] N. Le Sauze, A. Chiosi, R. Douville, H. Pouyllau, H. Lonsethagen, P. Fantini, C. Palas-ciano, A. Cimmino, M. A. Callejo Rodriguez, O. Dugeon, D. Kofman, X. Gadefait, P. Cuer, N. Ciulli, G. Carrozzo, A. Soppera, B. Briscoe, F. Bornstaedt, M. Andreou, G. Stamoulis, C. Courcoubetis, P. Reichl, I. Gojmerac, J. L. Rougier, S. Vaton, D. Barth, and A. Orda. ETICS : QoS-enabled interconnection for Future Internet services. In *Future Network and Mobile Summit*, 2010.

[20] TELECOM Bretagne Equipe SERES. ORBAC : Organization Based Access Control. http://orbac.org/index.php?page=home&lang=fr.

Annexe A : contributions en communications numériques

1. BERROU Claude, VATON Sandrine, JEZEQUEL Michel, DOUILLARD Catherine Computing the minimum distance of linear codes by the error impulse method.. IEEE GLOBECOM 2002, Taipei, Taïwan, 2002
2. VATON Sandrine, CHONAVEL Thierry, SAOUDI Samir Iterative multi-user detection and decoding for turbo-coded DS-CDMA systems.. Eusipco 2002 (XI European signal processing), 03-06 september 2002, Toulouse, France, 2002

Computing the Minimum Distance of Linear Codes by the Error Impulse Method

Claude Berrou, Sandrine Vaton, Michel Jézéquel and Catherine Douillard
Ecole Nationale Supérieure des Télécommunications
BP 832, 29285 Brest Cédex, France

Abstract - A new method for computing the minimum distances of linear error-correcting codes is proposed and justified. Unlike classical techniques that rely on exhaustive or partial enumeration of codewords, this new method is based on the ability of the Soft-In decoder to overcome Error Impulse input patterns. It is shown that the maximum magnitude of the Error Impulse that can be corrected by the decoder is directly related to the minimum distance. This leads to a very fast algorithm to obtain minimum distances of any linear code whatever the block size and the code rate considered. In particular, the method can be advantageously worked out for turbo-like concatenated codes.

Keywords: minimum distance, error floor, linear code, convolutional, concatenated, turbo code.

1 Introduction

The asymptotic performance of a linear block error correcting code at very low error rates is completely determined by its minimum distance $d_{\min}$ and by the multiplicity $n(d_{\min})$ if one considers the Frame Error Ratio (FER).

Consider a $C(n,k)$ linear block code (k is the number of information bits, n is the size of codewords). Denote by $d_{\min}$ its minimum distance; for any distance $d \geq d_{\min}$, denote by $n(d)$ its multiplicity (number of codewords with weight d). On the Gaussian channel with maximum-likelihood (ML) decoding the Frame Error Ratio (FER) can be upperbounded by the union bound:

$$\mathrm{FER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} n(d) \mathrm{erfc}((dR\frac{Eb}{N0})^{1/2}) \qquad (1)$$

where $\mathrm{erfc}(x)$ is the complementary error function: $\mathrm{erfc}(x) = 2/\sqrt{\pi} \int_x^{\infty} \exp(-t^2)\, dt$.

At low error rates, this upper bound is very tight; moreover, as the coefficients $\mathrm{erfc}(x^{1/2})$ decrease exponentially with x, the FER can be approximated, at low error rates, by the first term of the union bound:

$$\mathrm{FER} \simeq \frac{1}{2} n(d_{\min}) \mathrm{erfc}((d_{\min} R\frac{Eb}{N0})^{1/2}) \qquad (2)$$

When designing a code that operates at very low error rates, it is very difficult to determine its performance by Monte Carlo simulations. It is highly desirable to propose some techniques to determine the minimum distance $d_{\min}$ and its multiplicity $n(d_{\min})$.

For algebraic codes such as BCH codes or Reed Solomon codes, $d_{\min}$ is a parameter that is specified prior to the design of the code. For elementary convolutional codes, the Viterbi algorithm [4] can be used to estimate the minimum distance, but this algorithm cannot be used to estimate the minimum distance of concatenated codes and in particular of Turbo Codes (TCs).

It is then important to develop some fast algorithms to compute the minimum distance $d_{\min}$ of concatenated codes and, in particular, of TCs. This will make it possible to design efficient TCs without intensive computations. This will for example make it possible to design good component codes and good interleavers in record time.

The method that we propose in this paper is based on the notion of Error Impulse Response of the decoder. The Error Impulse Response of the iterative Soft-In/ Soft-Out (SISO) decoding algorithm was introduced in [2]. It has also been used in [5]. The proof of our method is based on a reasoning in the Euclidean space $\mathbb{R}^n$. For the proof, we suppose that the decoder is a maximum likelihood (ML) decoder on the Gaussian channel and that the modulation is BPSK or QPSK. In practice, the turbo decoder on the Gaussian channel is used.

2 Principle of the method

We use the following notations:

- $x = (-1, -1, \cdots, -1)$ is the word associated with the "all zero" codeword by the modulation.

This work will be presented at ISIT'02, Lausanne, Switzerland, July 2002.

0-7803-7632-3/02/$17.00 ©2002 IEEE

- $y = (-1, -1, \cdots, -1, -1 + A_i, -1, \cdots, -1)$ is the input to the decoder. A_i is a positive real number called the error impulse. i is the position of error.

We identify any codeword with the sequence of +1s and −1s. The Hamming weight of a codeword is the number of +1's in this codeword.

y is decoded according to the ML criterion on the Gaussian channel. The decoded codeword $\hat{x}$ is such that:

$$< \hat{x}, y > \geq < z, y > \qquad \forall z \in \mathcal{C} \qquad (3)$$

where $< \bullet, \bullet >$ is the scalar product and $\mathcal{C}$ is the code.

Theorem 1 *If there exists a position of error i and an error impulse A_i such that the decoded codeword is not the "all zero" codeword, then*

$$A_i \geq \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \qquad (4)$$

where the minimum is taken over all codewords z such that $z_i = +1$ and where $w_H(z)$ is the Hamming weight of z.

Proof:
Assume that $\hat{x} \neq x$. It results from Eq. (3) that:

$$< \hat{x}, y > \geq < x, y > \qquad (5)$$

$< x, y >= n - A_i$. Denote by w the Hamming weight of $\hat{x}$: $w = w_H(\hat{x})$. If $\hat{x}_i = -1$ then $< \hat{x}, y >= n - 2w - A_i$ and this is conflicting. Therefore $\hat{x}_i = +1$, $< \hat{x}, y >= n - 2w + A_i$ and it results from Eq. (5) that :

$$n - 2w + A_i \geq n - A_i \quad \Rightarrow \quad A_i \geq w \qquad (6)$$

Since $w \geq \min_{z \in \mathcal{C}/z_i = +1} w_H(z)$ we obtain

$$\min_{z \in \mathcal{C}/z_i = +1} w_H(z) \leq w \leq A_i \qquad (7)$$

Theorem 2 *If there exists a position of error i and a positive real number A_i such that the decoded codeword is the "all zero" codeword, then*

$$A_i \leq \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \qquad (8)$$

Proof

Assume that $\hat{x} = x$. It results from Eq. (3) that:

$$n - A_i \geq < z, y > \qquad \forall z \in \mathcal{C} \qquad (9)$$

Consider a particular z such that $z_i = +1$ and denote by $w = w_H(z)$ its Hamming weight. Then it results from Eq. (9) that :

$$n - A_i \geq n - 2w + A_i \quad \Rightarrow \quad A_i \leq w \qquad (10)$$

Therefore,

$$A_i \leq \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \qquad (11)$$

Theorem 3 *for any error position i, there exists a positive error impulse A_i^* such that*

$$\begin{aligned} A_i^* &= \min\{A_i \,/\, \hat{x} \neq x\} = \max\{A_i \,/\, \hat{x} = x\} \\ &= \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \end{aligned} \qquad (12)$$

The minimum distance of the code is :

$$d_{\min} = \min_i A_i^* \qquad (13)$$

Proof

It results from Th. (1) and (2) that

$$\max\{A_i \,/\, \hat{x} = x\} \leq \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \leq \min\{A_i \,/\, \hat{x} \neq x\} \qquad (14)$$

Suppose that $\max\{A_i \,/\, \hat{x} = x\} \neq \min\{A_i \,/\, \hat{x} \neq x\}$. There exists A_i such that $\max\{A_i \,/\, \hat{x} = x\} < A_i < \min\{A_i \,/\, \hat{x} \neq x\}$. For this A_i, either $\hat{x} \neq x$ which is contradictory to $A_i < \min\{A_i \,/\, \hat{x} \neq x\}$, or $\hat{x} = x$ which is contradictory to $A_i > \max\{A_i \,/\, \hat{x} = x\}$. Therefore, the inequalities in Eq. (14) are equalities and there exists A_i^* such that:

$$\begin{aligned} A_i^* &= \min\{A_i \,/\, \hat{x} \neq x\} = \max\{A_i \,/\, \hat{x} = x\} \\ &= \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \end{aligned} \qquad (15)$$

Any codeword $z \neq x$ has at least one position i such that $z_i = +1$. Therefore $d_{\min}$ is equal to

$$d_{\min} = \min_i \min_{z \in \mathcal{C}, z_i = +1} w_H(z) = \min_i A_i^* \qquad (16)$$

3 Algorithm

3.1 Estimation of the minimum distance as the minimum error impulse

We assume that $d_{\min}$ is in the range $[d_0, d_1]$ where d_0 and d_1 are two integers. Then $d_{\min}$ can be determined as follows.

- set $A_{\min} = d_1 + 0.5$.
- for i=1 to n do
  - $A = d_0 - 0.5$;
  - set $[(\hat{x} = x) = \text{TRUE}]$;
  - while $[(\hat{x} = x) = \text{TRUE}]$ and $(A \leq A_{\min} - 1.0)$ do
    - $A = A + 1.0$;
    - $y = (-1, \cdots, -1, -1 + A, -1, \cdots, -1)$ where $-1 + A$ is in position i;
    - ML decoding of y $\Rightarrow \hat{x}$;
    - if $(\hat{x} \neq x)$ then $[(\hat{x} = x) = \text{FALSE}]$;

    end while
  - $A_{\min} = A$

  end for
- $d_{\min}$ is the integer part of $A_{\min}$.

3.2 Practical issues

In most cases, the code is cyclic, which means that if a circular shift is applied to a codeword it is still a codeword. In this case, it is sufficient to test one position of error i to obtain the minimum distance. This is, for example, the case for convolutional codes if one neglects possible side effects due to termination.

For concatenated codes, when iterative SISO decoding is not available, the method cannot be applied because one-step decoding is largely suboptimal.

For turbo like (serially or parallel) concatenated codes the method can be applied with an iterative SISO decoder, although it has not been proved that turbo decoding is ML. This pattern of noise (no noise at all except in position i where A_i is large) is very improbable on the Gaussian channel. It is not sure that the turbo decoder maximizes $< y.z >$ since the quasi optimality of turbo decoding has been proven only by simulation with realistic patterns of noise.

Nevertheless the experiment shows good agreement between the minimum distances of TCs obtained with this algorithm and the asymptote obtained by simulation. The appropriate SISO decoder for each component code is the max-log-MAP algorithm [6]. This algorithm does not require the knowledge of the channel parameters. The extrinsic information should be passed from one component decoder to the other without any alteration (no attenuation, no saturation). The number of iterations needed for convergence depends on the expected value of $d_{\min}$. It may vary from a few tens to several hundreds for very large minimum distances. For example, in some cases, when $d_{\min}$ is around 20, the estimated $d_{\min}$ with 256 iterations is one point up on the value obtained with 128 iterations.

For the parallel concatenation of recursive systematic codes with interleaving, some restrictions should be made to the algorithm given in Section (3.1). In this case the error impulse A_i should be applied only on the systematic bits, but not on the parity bits. Parity bits are more sensitive than systematic bits to error impulses. Applying error impulses to parity bits would lead to underestimating $d_{\min}$.

When examining a turbo like concatenated code, the number of symbols to be tested is related to the periodicity properties of the code. In the case of a classical turbo code, this number, denoted T, is the lowest common multiple of the period of the interleaver and of the period of the puncturing pattern, if any. The former depends on the permutation model that was adopted in the design of the interleaver. When no periodicity is observable (for instance when the permutation function was obtained by some pragmatic computer-based research), T must be maximum (i.e. $T = k$).

Multiplicity of $d_{\min}$

Because the method does not explicitly provide the multiplicity of codewords of weight $d_{\min}$ and higher, we must introduce some further hypotheses to use the asymptote on the FER given in Eq. (2). These assumptions seem to be in good agreement with the performance obtained by simulation, although they are irrelevant for some other codes, such as product codes.

Hyp. 1: there is only one codeword z with weight A_i^* and such that $z_i = +1$.

Hyp. 2: all the distances A_i^* obtained for the whole set of positions i ($1 \leq i \leq k$) concern distinct codewords (there is no overlapping).

The former hypothesis is optimistic, while the latter is pessimistic. Both together, as confirmed in the examples given in the next Section, provide a good estimate of the FER, that can thus be calculated as:

$$\text{FER} \simeq \frac{1}{2} \sum_{i=1,k} \text{erfc}\left(\sqrt{RA_i^* \frac{Eb}{N0}}\right) \qquad (17)$$

Note that if the spectral thinning is pronounced the error floor will be completely determined by the minimum distance $d_{\min} = \min_i A_i^*$ and its multiplicity $n(d_{\min})$ which is estimated as the number of positions i for which $A_i^* = d_{\min}$.

4 Simulation results

4.1 DVB-RCS standard

This standard [3] uses duo-binary 8-state turbo codes. It was defined for various block sizes (12 to 206 bytes) and code rates ($R = 1/3$ to $6/7$). Figure 4.1 shows the performance in FER measured on FPGA hardware, for MPEG packets (188 bytes) and for rates 2/3 and 4/5. The asymptotic performance given by Eq. (2) is also displayed. The parameters obtained by the error impulse method are:

- $d_{\min} = 13$ and $n(d_{\min}) = 752$ for $R = 2/3$
- $d_{\min} = 8$ and $n(d_{\min}) = 940$ for $R = 4/5$

Measured and estimated curves are in good agreement.

4.2 UMTS/3GPP standard

A TC has been normalized in the UMTS/3GPP [1] standard for personal communications. This standard uses binary 8-state RSC codes as component codes. The error impulse method has been tested on this turbo code for a relatively small interleaver: $k = 640$ bits. In this case

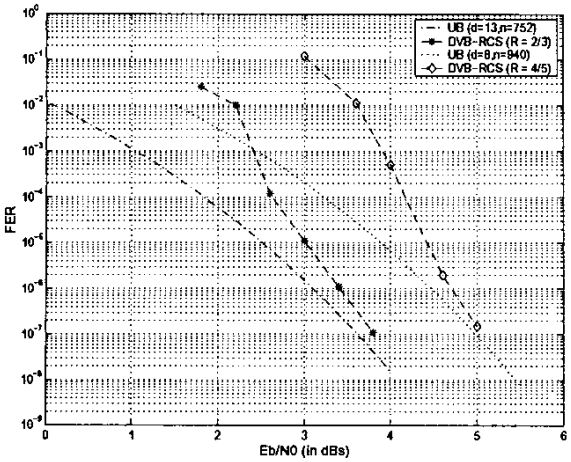


Figure 1: Frame Error Rate of the DVB-RCS turbo code for $k = 1504$ and rates 2/3 and 4/5. Both simulated FER and estimated FER (UB, eq. (2))) are given.

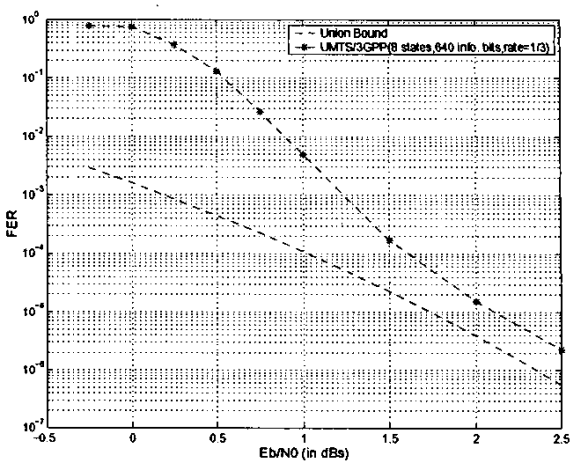


Figure 2: Frame Error Rate of the UMTS/3GPP turbo code for $k = 640$ and rate 1/3. Both simulated FER and estimated FER (UB, eq. (17)) are given.

the spectral thinning is not extremely pronounced because the interleaver size is moderate. Therefore, we have taken into account not only the codewords of weight $d_{\min}$ but also codewords with weights $d \geq d_{\min}$. Figure 4.2 shows the performance obtained by simulations and the asymptotic performance derived from Eq. (17). The "spectrum" provided by the error impulse method is the following:

$d_{\min} = 26$

$n(d), d = 26, 27, \cdots$				1	1	31	91	30	22
214	36	24	31	27	31	34	27	22	9
6	3	0	0	0	0	$\cdots$	$\cdots$		

where, this time, $n(d)$ denotes this time the number of positions i where the minimum error impulse is $A_i^* = d$.

There is good agreement between the performance simulated, and the performance estimated by the Error Impulse method. A loss of 0.2–0.3 dB can be observed; this loss has been observed by many contributors [2][7] and may be due to the suboptimality of the decoder.

5 Conclusion

A powerful tool for the design of linear codes, and in particular of turbo codes, has been introduced and justified. Whatever block sizes and coding rates, it gives the possibility to forecast performance at very low error rates in seconds or minutes. It may be used for instance as a fundamental algorithm in the search for good permutations in the construction of turbo codes. Nevertheless, further investigation has to be conducted to understand better the behaviour of turbo decoders facing Error Impulse sequences.

References

[1] 3rd Generation Partnership Project. Multiplexing and Channel Coding (FDD), June 1999. 3G TS 25.212.

[2] C.Berrou. Some clinical aspects of turbo codes. In *Proc. of the Intern. Symp. on Turbo Codes and Rel. Topics*, pages 26–31, Brest, France, Sep. 1997.

[3] ETSI. Digital Video Broadcasting (DVB). Interaction Channel for Satellite Distribution Systems, Feb. 2000. DVB-RCS001.

[4] G.D.Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61:268–278, march 1973.

[5] J.B.Anderson. Transient and convergence properties of the bcjr decoder via a linear system model. In *Proc. of the Information Theory Workshop*, pages 135–137, Cairns, Australia, Sep. 2001.

[6] P.Robertson, P.Hoeher, and E.Villebrun. Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding. *Europ. Trans. on Telecom.*, 8:119–125, March-Apr 1997.

[7] S.Benedetto, L.Gaggero, R.Garello, and G.Montorsi. On the design of binary serially concatenated convolutional codes. In *Proc. of the VIII Communication Theory Mini-Conf, CTMC*, pages 32–36, Vancouver,BC,Canada, June 1999.

Iterative multi-user detection and decoding for turbo-coded DS-CDMA systems

Sandrine Vaton, Thierry Chonavel and Samir Saoudi
ENST-Bretagne, BP 832, 29285 BREST Cedex, FRANCE
email: {sandrine.vaton, thierry.chonavel, samir.saoudi}@enst-bretagne.fr

ABSTRACT

In this paper we consider turbo-coded multi-user DS-CDMA communications. We first present a new scheme for Soft-Input Soft-Output multi-user detection. Our method provides the Maximum A Posteriori estimate for each user bit together with its fiability in the form of an extrinsic information. It is based on the stack algorithm over the tree diagram of the CDMA channel. The Soft-Input Soft-Output multi-user detector is then associated to a bank of turbo-decoders in a global turbo loop to perform iteratively multi-user detection and turbo-decoding. Simulation results show the high efficiency of the proposed method.

1 Introduction

The turbo principle was first introduced in [4] for the iterative decoding of parallel concatenated convolutional codes; it has been generalized to different communication problems such as the decoding of serially concatenated codes, joint source and channel coding, turbo interference reduction or turbo equalization.

Recently some authors have considered the application of the turbo principle to the separation and decoding of K coded users that communicate over a CDMA channel [8][2][10][12]. Basically, the solution consists in considering the CDMA channel as a code, and in applying the principles of the turbo decoding of serially concatenated convolutional codes [3].

The problem of Maximum Likelihood (ML) estimation of the information bits is broken into two "simple" ones: (i) Soft Input Soft Output (SISO) Multiple Access Interference suppression, and (ii) SISO decoding. The two modules communicate iteratively in a global loop by exchanging a "global" extrinsic information over information bits and coded bits, until no substantial amelioration is provided by an additional iteration. The SISO decoder itself is turbo since the code considered in this paper is the turbo code normalized in the UMTS [1]; the turbo code consists in the parallel association of two elementary 8-states convolutional codes. Therefore, in the global loop, one pass of the SISO decoder can be decomposed into several passes in the local turbo decoding loop. In the local loop a local extrinsic information over the information bits is exchanged between the two component SISO 8-states trellis decoders.

In Section 2, we present a suboptimal Maximum A Posteriori multiuser detector with soft output. Our subMAP detector is based on the stack algorithm [7] over the tree diagram of the CDMA channel. The stack algorithm presents two advantages (i) it provides the exact ML estimate and (ii) its numerical complexity is linear with the number of users when the Signal to Noise Ratio (SNR) is high. In Section 3 we integrate in the branch metrics of the tree diagram a Bayesian prior over the user bits; this Bayesian prior is in practice the global extrinsic information provided by the bank of turbo decoders. In Section 3 we also explain how one computes the extrinsic information provided by the SISO detector. In Section 4, we discuss the association of the multiuser detector with the bank of single user turbo decoders. Simulation results are presented in Section 5.

2 Multiuser detection based on the tree diagram

Recall that the model of a CDMA channel with additive white Gaussian noise is the following:

$$\mathbf{r} = \mathbf{S}\mathbf{b} + \mathbf{n}, \tag{1}$$

where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the vector of users bits, $\mathbf{S} = [\mathbf{s_1}, \ldots, \mathbf{s_K}]$ is the matrix of normalized signatures, and $\mathbf{n}$ is a zero mean Gaussian vector with independent components: $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$.

Clearly, the problem of ML detection on the additive white Gaussian noise channel is equivalent to minimizing $\| \mathbf{Sb} - \mathbf{r} \|^2$:

$$\hat{\mathbf{b}} = \mathrm{Arg} \min_{\mathbf{b}\in\{\pm 1\}^{\mathbf{K}}} \| \mathbf{Sb} - \mathbf{r} \|^2 . \tag{2}$$

Consider the QR decomposition of $\mathbf{S}$: $\mathbf{S} = \mathbf{QR}$ where $\mathbf{R}$ is an upper triangular matrix and where $\mathbf{Q}$ is a unitary matrix, that is to say $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. It can be shown that minimizing $\| \mathbf{Sb} - \mathbf{r} \|^2$ amounts to minimizing $\| \mathbf{Rb} - \mathbf{z} \|^2$, where $\mathbf{z} = \mathbf{Q^T r}$ is an exhaustive statistics for the problem. The upper triangular structure of $\mathbf{R}$ makes it possible to solve this problem as the detection of a shortest path in a tree diagram of the form displayed on Figure 1. The branch metrics over the tree diagram are:

$$\text{branch } (b_K, b_{K-1}, \cdots, b_k) \leftarrow [(\mathbf{Rb})_k - z_k]^2 \tag{3}$$

where $(\mathbf{Rb})_k = \sum_{i=k,K} R(k,i)b_i$ is the component number k of vector $\mathbf{Rb}$.

Recently, we have proposed to use the stack algorithm to solve this problem [13]. The stack algorithm (Zigangirov, 1966 [14]; Jelinek, 1969 [7]) keeps track of a few paths and their corresponding metrics in a stack; the head of the stack is always the path with the shortest metric among all the paths in the stack. At each step of the algorithm, the path at the head of the stack is extended by one branch, thus yielding two successors; the two successors along with the other paths in the stack are reordered, so that the head of the stack is always the path with the shortest metric. The process of extending the path is then repeated until a complete path with K branchs is found.

The stack algorithm has the advantage to produce the exact ML solution, contrary to the other methods over the tree diagram such as the Fano algorithm or the feedback decoding algorithm which are suboptimal [9]. What is more, for high SNRs, the number of branch metrics computed per user is equal to 2, leading to a linear complexity with the number of users. At a high SNRs, the stack algorithm never turns back over the tree diagram; the process of selecting the best successor among the two successors of the head of the stack always selects the true path because this path is much more probable that the other paths in the stack.

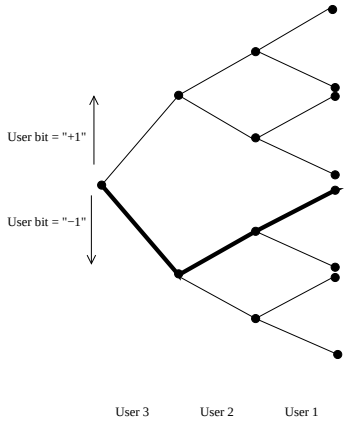


Figure 1: Tree diagram for $K = 3$ users. The bold path is $\mathbf{b} = (b_1, b_2, b_3)^T = (+1, +1, -1)^T$.

3 SubMAP detection over the tree diagram

In order to perform iteratively the multiuser detection and the decoding in a turbo manner, we must adapt the detection algorithm (i) to account for a Bayesian prior over the user bits (this information is supplied by the output of the turbo decoders at the previous iteration) (ii) and to supply a soft decision in form of an extrinsic information that will be used as the input of the bank of turbo-decoders.

3.1 Branch metrics update with Bayesian prior

Suppose that there is a Bayesian prior over b_k:

$$\log p(b_k = \pm 1; Z_k^{\text{in}}) = \text{sign}(b_k)\ Z_k^{\text{in}} + \text{constant}, \qquad (4)$$

Then the complete log-likelihood of $\log p(\mathbf{b} \mid \mathbf{z}; \mathbf{Z}^{\text{in}})$ is, up to an additive constant term:

$$\begin{aligned} \log p(\mathbf{b} \mid \mathbf{z}; \mathbf{Z}^{\text{in}}) \quad = \quad & \textstyle\sum_{k=K}^{1} \text{sign}(b_k)\ Z_k^{\text{in}} \\ & -(2\sigma^2)^{-1} \textstyle\sum_{k=K}^{1} ((\mathbf{Rb})_k - z_k)^2 \end{aligned} \qquad (5)$$

This is the cumulated metrics of the path $\mathbf{b}$ where the branch metrics are defined as:

$$\text{branch } (b_K, b_{K-1}, \cdots, b_k) \leftarrow \text{sign}(b_k) Z_k^{\text{in}} - \frac{1}{2\sigma^2}[(\mathbf{Rb})_k - z_k]^2. \qquad (6)$$

where $\mathbf{Z}^{\text{in}} = (Z_1^{\text{in}}, \cdots, Z_K^{\text{in}})$ is the parameter of the distributions.

3.2 Extrinsic information computation

The a posteriori log-likelihood ratio of b_k is:

$$\text{LLR}(b_k \mid \mathbf{z}) = \log \frac{p(b_k = +1 \mid \mathbf{z}; \mathbf{Z}^{\text{in}})}{p(b_k = -1 \mid \mathbf{z}; \mathbf{Z}^{\text{in}})}. \qquad (7)$$

For computational reasons it is approximated as:

$$\begin{aligned} \text{LLR}(b_k \mid \mathbf{z}; \mathbf{Z}^{\text{in}}) \quad & \simeq \max_{\mathbf{b}/b_k=+1} \log p(\mathbf{b}, \mathbf{z}; \mathbf{Z}^{\text{in}}) \\ & - \max_{\mathbf{b}/b_k=-1} \log p(\mathbf{b}, \mathbf{z}; \mathbf{Z}^{\text{in}}). \end{aligned} \qquad (8)$$

This suboptimal approach considers only the best path such that $b_k = +1$, and the best path such that $b_k = -1$. This approximation is valid at high SNRs. It is conventional to use suboptimal components in turbo loops since the iterations have the potentiality to correct these suboptimalities [11]. Note that in Eq. (8) the joint log-likelihood $\log p(\mathbf{b}, \mathbf{z})$ is the cumulated metrics of the path $\mathbf{b}$ in the tree diagram with the branch metrics defined by (6).

Suppose first of all that $k = K$, that is to say that the user considered corresponds to the root of the tree displayed on Figure 1. The half upper part of the tree corresponds to $b_K = +1$ whereas the half lower part of the tree corresponds to $b_K = -1$.

The most probable path such that $b_K = +1$ is then obtained by application of the stack algorithm to the tree reduced to its half upper part. This reduction can be performed, for example, by setting the metrics of the branch $b_K = -1$ to minus infinity. Similarly, the most probable path such that $b_K = +1$ is obtained with the stack algorithm over the tree reduced to its half lower part.

Suppose now that $k < K$ so that the user k is not at the root of the tree. We propose the following strategy: first, we reorder the users in vector $\mathbf{b}$ so that bit b_k be at the root; after reordering,

$$\mathbf{b} = (b_1, \cdots, b_{k-1}, b_{k+1}, \cdots, b_K, b_k)^T. \qquad (9)$$

The same permutation should be applied on the vectors $\mathbf{z}$ and $\mathbf{Z}^{\text{in}}$ as well as on the columns of the matrix $\mathbf{R}$.

The permutation of the columns of $\mathbf{R}$ produces one subdiagonal element on columns $k, k+1, \cdots, K-1$. With subdiagonal elements the tree representation of the problem is lost.

The subdiagonal elements are therefore eliminated by means of Givens rotations [6]. The first rotation combines the line $K-1$ and the line K of the matrix $\mathbf{R}$ to eliminate the subdiagonal element $R(K,K-1)$ of column $K-1$. The second rotation combines the lines $K-2$ and $K-1$ of the matrix $\mathbf{R}$ to eliminate the subdiagonal element $R(K-1,K-2)$ on column $K-2$, and so on. The same Givens rotations are applied to the vector $\mathbf{z}$.

Givens rotations are unitary transforms. Therefore, the quantity $||\mathbf{z}-\mathbf{Rb}|\,|^2$ is not modified by the set of permutations and Givens rotations. The complete log-likelihood $\log p(\mathbf{b},\mathbf{z};\mathbf{Z}^{\text{in}})$ is the cumulated metrics of the path $\mathbf{b}$ in the reordered tree diagram. And for b_k the computation of the a posteriori log-likelihood ratio (7) can be done easily from the reordered tree diagram.

The global extrinsic information produced by the SISO detector is obtained as:

$$Z_k^{\text{out}} = 0.5\ \text{LLR}(b_k \mid \mathbf{z};\mathbf{Z}^{\text{in}}) - Z_k^{\text{in}}. \tag{10}$$

The extrinsic information represents the distribution of bit b_k given the soft channel values $\mathbf{z}$, and given the Bayesian priors over bits $b_i, i \neq k$ with an equiprobable prior over bit b_k:

$$Z_k^{\text{out}} = 0.5\ \log \frac{p(b_k=+1 \mid \mathbf{z};\mathbf{Z}_{-k}^{\text{in}})}{p(b_k=-1 \mid \mathbf{z};\mathbf{Z}_{-k}^{\text{in}})}. \tag{11}$$

where $\mathbf{Z}_{-k}^{\text{in}} = (Z_1^{\text{in}},\cdots,Z_{k-1}^{\text{in}},0,Z_{k+1}^{\text{in}},\cdots,Z_K^{\text{in}})$. As always, the Bayesian prior Z_k^{in} over b_k is not taken into account in the output extrinsic information Z_k^{out} of the same bit. This makes it possible to avoid too strong correlations between the successive iterations of the global turbo process. Too much correlation between the iterations could indeed lead to a fixed point that could be a local extremum of the likelihood.

4 Turbo decoding and the global extrinsic information passing

In a very similar way to the turbo decoding of serially concatenated codes, the extrinsic informations Z_k^{out} produced by the detector are the only inputs of the bank of turbo decoders. These quantities are demultiplexed at the output of the detector, so that each turbo decoder has as inputs the extrinsic informations relative to the information bits and parities of the concerned user (see Fig. 3).

The turbo decoding is performed as if the extrinsic informations Z_k^{out} were soft channel values over an additive white Gaussian noise channel. Indeed if $z = b+n$ where $b=\pm 1$ and where n is a Gaussian random variable with mean 0 and variance σ^2, then $\log[p(b_k=+1 \mid z)/p(b_k=-1 \mid z)] = 2/\sigma^2 z$. z is therefore homogeneous to half the log-likelihood ratio. The turbo decoding loop is displayed on Fig. 2. Each component decoder generates a local extrinsic information relatively to the information bits only. This extrinsic information is exchanged iteratively between the two component SISO decoders inside the local loop. At the last iteration of turbo decoding, the turbo decoder provides a global extrinsic information relatively to both the information bits and the parities. This information is globally extrinsic because, if it concerns bit b_k, it does not take into account the input of the turbo decoder relatively to b_k but only the inputs relatively to bits $b_i, i \neq k$. The global extrinsic information produced by the turbo decoder is then one of the inputs of the detector, together with the channel values, at the next iteration of the global loop.

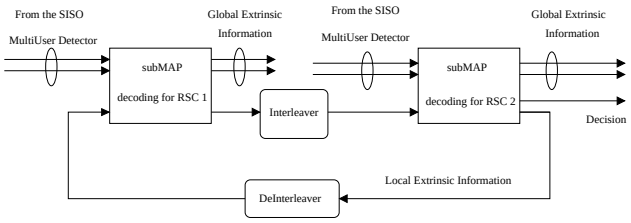


Figure 2: Local Extrinsic Information passing.

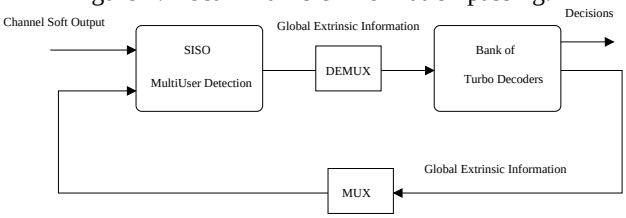


Figure 3: Global Extrinsic Information passing.

5 Simulation results

For computational issues we consider the case of $K=9$ users with $N_c=20$ chips. The signatures are Gold sequences of length 63, truncated at $N_c=20$.

5.1 Tree decoding algorithm

We first simulate the performances of the (hard output) ML detector based on the stack algorithm over the tree diagram. Figure 4 plots the BER obtained with the stack algorithm, the feedback decoding algorithm [13], which implements approximate ML detection by means of a sliding window of length L (here, $L=3, L=2, L=1$), the decorrelator detector, as well as the single user bound (BPSK). At high SNRs the stack algorithm achieves the single user bound, unlike other techniques. For example, a gain of 3.57 dB is obtained at a BER of 10^{-5} if one replaces the standard decorrelator by the stack algorithm. The average number of branch metrics computed to go through the tree is displayed on Figure 5 for different SNRs. At high SNRs the number of branch metrics computed is equal to 2 per user.

5.2 Iterative SISO detection/turbo-decoding

We consider now the iterative association of the subMAP multiuser detector to a bank of turbo decoders as displayed on Fig. 3. Within each global loop, 5 iterations of turbo decoding are performed. Moreover, we consider up to 5 global iterations. The normalized UMTS turbo code with 640 information bits is used; this turbo code has a free distance $d_f=26$ [5]. In the decoding the SNR is not known and

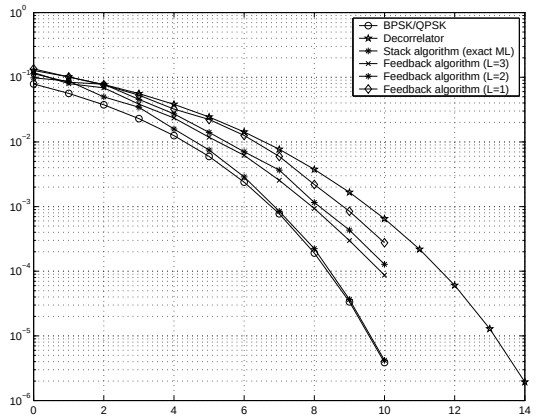


Figure 4: Stack algorithm, feedback decoding algorithm ($L = 3$,$L = 2$ and $L = 1$), decorrelator. $K = 9$ users, Gold sequences of length 63 truncated at $N_c = 20$.

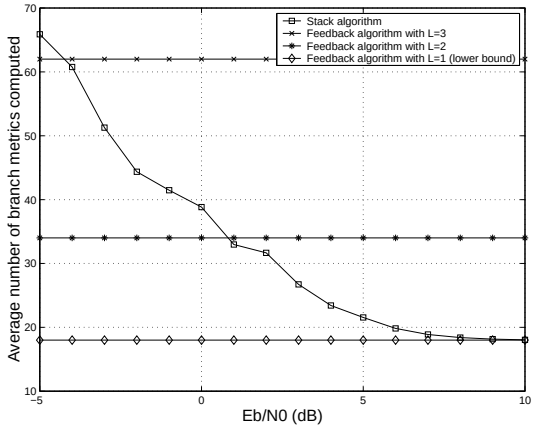


Figure 5: Average number of branch metrics computed per block of K users. $K = 9$ users, Gold sequences of length 63 truncated at $N_c = 20$.

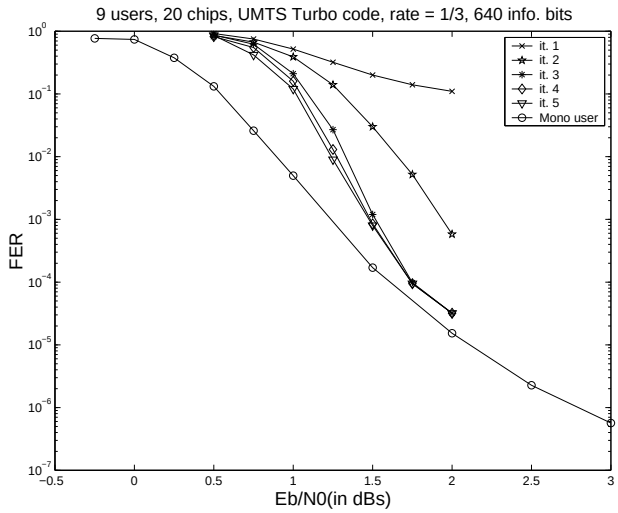


Figure 6: $K = 9$ users, $N_c = 20$ chips, UMTS Turbo Code (rate = 1/3), 640 information bits per user, 5 iterations of turbo decoding per global iteration, 5 global iterations.

we set $\sigma^2 = 1$ in the branch metrics. The SISO component decoders are max log MAP decoders [11]. The Frame Error Rate after each global iteration is displayed on Fig. 6. Convergence is obtained after only 3 iterations; this can easily be explained by the quasi optimality of the subMAP detector, and of the turbo decoder (with 5 iterations). A residual gap of about 0.1 dB is observed between the single user bound and the multi user performance.

6 Conclusion

In this paper we have proposed a new SISO multiuser detection structure of the subMAP type, and we have associated it to a turbo-decoder in an iterative global detection/decoding loop. We have checked by simulations the high efficiency of the proposed structure.

References

[1] Spreading and modulation (FDD). Technical report, Third Generation Partnership Project (3GPP), TS 25.213 v2.3.0, 1999-9.

[2] P. Alexander, M. Reed, J. Asenstorfer, and C. Schlegel. Iterative multiuser interference reduction : Turbo CDMA. *IEEE Trans. on Comm.*, 47:1008–1014, july 1999.

[3] S. Benedetto and G. Montorsi. Iterative decoding of serially concatenated convolutional codes. *Electronic Letters*, 32:1186–1188, june 1996.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding : Turbo Codes. In *Int. Conf. on Comm.*, pages 1064–1070, 1993.

[5] C. Berrou and S. Vaton. Computing the minimum distance of linear codes by the error impulse method. In *Submitted to ITC 2002.*

[6] G. H. Golub and C. F. V. Loan. *Matrix Computation - Second Edition.* John Hopkins Series in Mathematical Sciences, John Hopkins University Press, 1990.

[7] F. Jelinek. Fast sequential decoding algorithm using a stack. *IBM Jour. Res. Dev.*, 13:675–685, November 1969.

[8] M. Moher. An iterative multiuser decoder for near-capacity communications. *IEEE Trans. on Comm.*, 46:870–880, july 1998.

[9] J. Proakis. *Digital Communications.* McGraw-Hill Higher Education, 2001.

[10] Z. Qin, K. Teh, and E. Gunawan. Iterative multiuser detection for asynchronous CDMA with concatenated convolutional coding. *IEEE Journ. on Selected Areas in Comm.*, 19(9):1784–1791, 2001.

[11] P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In *Int. Conf. on Comm.*, pages 1009–1013, Seattle, june 1995.

[12] A. Tarable, G. Montorsi, and S. Benedetto. A linear front end for iterative soft interference cancellation and decoding in coded CDMA. In *IEEE Int. Conf. on Comm.*, volume 1, pages 1–5, 2001.

[13] S. Vaton, T. Chonavel, and S. Saoudi. Approximate and exact ML detectors for CDMA and MIMO systems: a tree detection approach. *Submitted to IEEE Comm. Letters.*

[14] K. Zigangirov. Some sequential decoding procedures. *Probl. Peredach. Inform.*, 2:13–25, 1966.

Annexe B : autres contributions en modélisation statistique de trafic et de réseaux

1. VATON Sandrine, CHONAVEL Thierry, Estimating the offset parameters of a mixture in the Fourier domain. ICASSP'99 (IEEE international conference on acoustics, speech and signal processing).Phoenix, March 15-19, 1999, pp. 1553-1556
2. SALAMATIAN Kave, VATON Sandrine, Hidden Markov modelling for network communication channels. ACM Sigmetrics 2001 Conference, Cambridge, Massachussets, USA, 2001
3. FILLATRE Lionel, MARAKOV Dmitry, VATON Sandrine, Forecasting Seasonal Traffic Flows with SARIMA Models, EuroNGI Workshop on QoS and Traffic Control, Ecole Normale Supérieure, Paris, décembre 2005

ESTIMATING THE OFFSET PARAMETERS OF A MIXTURE IN THE FOURIER DOMAIN

S. Vaton and T. Chonavel

ENST Bretagne, BP 832,29285 Brest Cedex, France.
e-mail: Sandrine.Vaton, Thierry.Chonavel@enst-bretagne.fr

ABSTRACT

In this contribution we present an algorithm for estimating some parameters of offset in the case of incomplete data. This estimation cannot be performed directly with an EM or SEM method because the density of local extrema in the likelihood map grows exponentially with the number of observations and because the SEM method provides a monotonic sequence of estimates so that bad initialization cannot be recovered. We perform the estimation in the Fourier domain. The offsets in time domain are transformed into pulsations in the Fourier domain. We minimize a quadratic distance between the parametric and empirical sampled Fourier transform with an EM method. Contrary to the problems encountered in the time domain the asymptotic loglikelihood of the sampled empirical Fourier transform is continuous w.r.t. the parameters of offset. We discuss the influence of the frequencies at which the Fourier transform is sampled and we present a numerical study of convergence of the proposed algorithms.

1. INTRODUCTION

Some previous studies have proved that Fourier transform based techniques enable estimation of the proportions of a mixture (see [1] and the references therein) . In this paper we discuss the possibility of using weighted distances between Fourier transforms and their empirical counterparts for estimating some parameters of offset in the case of incomplete data, a problem that cannot be solved by classical approaches. Semi-Markov processes (HMM, MRP...) with shifted exponential conditional laws $f_{\theta,\lambda}(x) = \lambda \exp(-\lambda(x-\theta))\mathbb{I}_{[\theta,+\infty[}(x)$ are generalizations of standard models of teletraffic such as the Poisson process or the MMPP. These models with offset fit the marginal distribution of the inter-event times better than standard models [2]. Modeling and estimating teletraffic is an important issue for dimensioning telecommunication networks.

The rest of the paper is organised as follows. In Section 2 we explain why MLE in the time domain is not viable in the case of incomplete data. In Section 3 we recall some techniques for estimating a single offset and we note that these techniques do not generalize to the case of incomplete data. In order to overcome these problems we present two Fourier transform based algorithms. The influence of the points at which the Fourier transform is sampled and the performances of the algorithm is discussed.

This work was supported by France Telecom research center under contract number PE95-7633.

2. FAILURE OF MLE IN THE TIME DOMAIN

2.1. Failure of the EM algorithm

Denote by $L(x;\theta)$ the loglikelihood of the observations $x = x_{1:T}$ when the parameters of shift are equal to $\theta = (\theta_i)_{1\leq i\leq K}$. When $(X_t)_t$ is an i.i.d. sequence distributed as a mixture $L(x;\theta) = \sum_t \log(\sum_i \pi_i f_{0,\lambda_i}(x_t - \theta_i))$ and $L(x;\theta)$ is discontinuous at any point $\theta = (\theta_i)_{1\leq i\leq K}$ such that one of the x_t is equal to one of the θ_i. Consequently the likelihood map $L(x;\theta)$ has infinitely many local extrema that attract the EM algorithm [3]. When $(X_t)_t$ is a HMM same discontinuities are encountered.

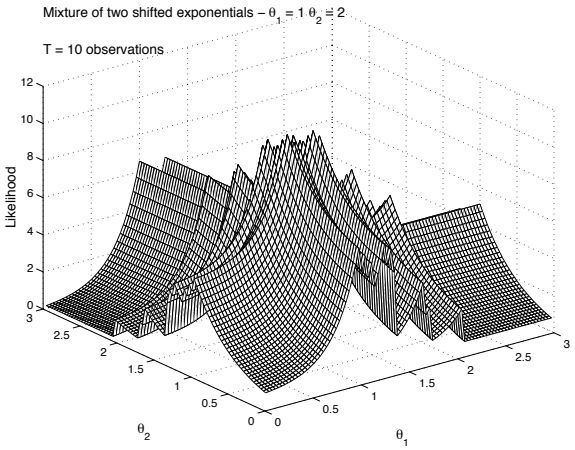


Figure 1: Discontinuities of the likelihood in time domain. $\pi_1 = \pi_2 = 0.5, \theta_1 = 1, \theta_2 = 2, \lambda_1 = 2, \lambda_2 = 4$.

2.2. Failure of the SEM algorithm

In the presence of local extrema one often uses the SEM algorithm [4]. Each iteration of the SEM algorithm can be decomposed into two steps:

1. Stochastic Expectation step

Compute by means of the forward backward algorithm the distribution of the unobserved $S_{1:T}$ conditionally to the observed $X_{1:T} = x_{1:T}$ for the value θ^k of the set of parameters. Simulate $S_{1:T}$ under this distribution. The result is denoted by $s^k = s^k_{1:T}$.

2. Maximization step

Maximize the complete loglikelihood $L(x_{1:T}, s^k_{1:T}; \theta)$:

$$L(x_{1:T}, s^k_{1:T}; \theta) \quad = \quad L(s^k_{1:T}) + \sum_i \sum_t \mathbb{1}_{s^k_t = i} \log f_{\theta_i, \lambda_i}(x_t)$$

If $x_t < \theta^k_i$ then $\mathbb{P}(s_t = i \mid x_t; \theta^k) = 0$ and a.s. $s^k_t \neq i$ so that $\theta^k_i \leq \min(x_t; s^k_t = i)$. If $x_t < \theta_i$ then $\sum_t \mathbb{1}_{s^k_t = i} \log f_{\theta_i, \lambda_i}(x_t) = -\infty$ so that $\theta^{k+1}_i \leq \min_{1 \leq t \leq T; s^k_t = i}(x_t)$ and this function strictly increases over $]-\infty, \min(x_t; s^k_t = i)[$ yielding $\theta^{k+1}_i = \min(x_t; s^k_t = i) \geq \theta^k_i$.

The sequence $(\theta^k_i)_{k \in \mathbb{N}}$ increases so that a bad initialization cannot be corrected.

2.3. Lack of Cramer-Rao Lower Bound (CRLB)

Maximum likelihood techniques are justified by the fact that the MLE is asymptotically unbiased and that its variance converges to $CRLB = \mathbb{E}_\theta(\frac{d^2\theta}{d\theta^2} \log f(X; \theta))$. For the problem under study the loglikelihood is not derivable w.r.t. the parameters of shift and the conditions under which the CRLB is derived are not fulfilled. However, in the Fourier domain the normalized sampled Fourier transform converges to a gaussian distribution which belongs to the exponential family $f(x; \theta) = c(\theta)h(x) \exp(< \alpha(\theta), T(x) >)$ and the MLE is known in this case to be efficient.

3. ESTIMATION OF A SINGLE SHIFT PARAMETER

In this Section we recall some techniques for estimating a parameter of offset when the observations are i.i.d. with p.d.f. $f_\theta(x) = f_0(x - \theta)$ and $f_0(x) = 0$ if $x < 0$.

3.1. Maximum Likelihood Estimation

The loglikelihood of $x_{1:T}$ is $L(x_{1:T}; \theta) = \sum_{t=1}^T \log f_0(x_t - \theta)$ so that $L(x_{1:T}; \theta) = -\infty$ if $x_t < \theta$ for some t. In the case when $f_0(\bullet)$ strictly decreases over $\mathbb{R}^+$ and in particular in the case of the shifted exponential distribution $\hat{\theta}_{ML} = X_{\min} \triangleq \min_{1 \leq t \leq T} X_t$.

3.2. Bayesian estimation techniques

The choice of a prior distribution is usually the controversial point in Bayesian estimation. Denote by $g(\theta)$ the prior distribution of θ. Then $f(x_{1:T}) = (f_0 * g)(x_{1:T})$ where $*$ denotes convolution. $f(x_{1:T})$ is maximum for $g(\theta) = \frac{f_0(x_{1:T} - \theta)}{\int f_0(x_{1:T} - \theta)\, d\theta}$ which acts as a matched filter on $f_0(x_{1:T})$. For this choice of prior distribution the Maximum A Posteriori is $\hat{\theta}_{MAP} = Arg \max_\theta f(\theta \mid x_{1:T}) = X_{\min}$ when f_0 strictly decreases over $\mathbb{R}^+$ and the mean squared error estimator is $\hat{\theta}_{EQM} = \mathbb{E}(\theta \mid X_{1:T}) = X_{\min} - (2\lambda T)^{-1}$ in the case of the shifted exponential distribution.

3.3. Pitman estimator

Consider the set of all equivariant estimators of θ, $S(x_{1:T} + \mu) = \mu + S(x_{1:T})$. It is natural to look for an estimator satisfying this property when one is concerned with the estimation of a parameter of shift. The equivariant estimator with minimum quadratic risk function is the Pitman estimator [5]. In the case of the shifted exponential distribution $\hat{\theta}_{PIT} = X_{\min} - (\lambda T)^{-1}$.

3.4. Barankin estimator

The Barankin bound [6] is the greatest lower bound among minimum variance bounds for unbiased estimators. As a byproduct the calculation of this bound supplies a locally, and possibly globally, minimum variance unbiased estimate. In particular, for a shifted exponential distribution, we can obtain the minimum variance unbiased estimator $\hat{\theta}_B(X) = X_{\min} - (\lambda T)^{-1}$ [6].

The above methods do not permit estimation of the offset parameters of a mixture. In order to overcome the limitations of these methods we propose solving this problem in the Fourier transform domain.

4. ESTIMATION IN THE FOURIER DOMAIN

4.1. The offsets in the time domain are transposed into pulsations in the Fourier domain

Denote by $\Phi(\omega) = \int e^{i\omega x} f(x) dx$ the characteristic function of f(x). Note that a shift in the time domain is equivalent to a modulation by a complex exponential in the Fourier domain the offset parameters $(\theta_i)_{1 \leq i \leq K}$ being the frequencies of the modulating complex exponentials : $\Phi(\omega) = \sum_i \pi_i e^{i\omega\theta_i} \Phi_{0,\lambda_i}(\omega)$. This note will permit the construction of an estimator of $\theta = (\theta_i)_{1 \leq i \leq K}$ from a sampled empirical estimate $(\hat{\Phi}(\omega_1), \cdots, \hat{\Phi}(\omega_L))$ where $\hat{\Phi}(\omega) = \frac{1}{T} \sum_{1 \leq t \leq T} e^{i\omega X_t}$.

4.2. Central Limit Theorem [7]

Denote by $\omega_1, \omega_2, \cdots, \omega_L$ the pulsations at which the Fourier transform is sampled. Denote by $m(\theta) = (\Phi(\omega_1), \Phi(\omega_2), \cdots, \Phi(\omega_L))^T$ the sampled Fourier transform of $f(\bullet)$ and denote the empirical sampled Fourier transform by $Z_T = T^{-1} \sum Y_t$ where Y_t is the vector with entry $1 \leq l \leq L$ is $e^{i\omega_l X_t}$.

Theorem 1 *Suppose that $(X_t)_t$ is i.i.d. or a finite state irreducible Markov chain; then it holds that $\sqrt{T}(Z_T - m(\theta)) \sim \mathcal{AN}(0, \Gamma(\theta))$ where $\Gamma(\theta) = \mathbb{E}_\theta((Y_t - m)(Y_t - m)^H)$ in the i.i.d. case and $\Gamma_\theta = \sum_{\tau \in \mathbb{Z}} \mathbb{E}_\theta((Y_{t+\tau} - m)(Y_{t+\tau} - m)^H)$ in the HMM case.*

4.3. Maximum Likelihood Estimation

Our procedure consists in minimizing the asymptotic loglikelihood of the normalized sampled Fourier transform :

$$J(\theta) = \log |\Gamma(\theta)| + \frac{1}{2} T (Z_T - m(\theta))^H \Gamma^{-1}(\theta)(Z_T - m(\theta))$$

Standard optimization techniques (gradient method, Newton method, conjugate gradient method ...) can be used to maximize $J(\theta)$. Let us consider the asymptotic covariance matrix $\Gamma(\theta)$. When $(X_t)_t$ is i.i.d. then $\Gamma_{ij}(\theta) = \Phi_\theta(\omega_i - \omega_j)$. When $(X_t)_t$ is a HMM it is simpler to replace $\Gamma(\theta)$ with a consistent estimate $\hat{\Gamma}_T$ and to optimize $K(\theta) = (Z_T - m(\theta))^H \hat{\Gamma}_T^{-1} (Z_T - m(\theta))$.

4.3.1. Problems related to sampling the Fourier transform

Contraction of the loglikelihood map

Suppose that $\omega_1 < \omega_2 < \cdots < \omega_L$. If $\omega_1 \leq 2\pi / X_{\max}$ where $X_{\max} = \max_{1 \leq t \leq T} X_t$ there exists a one to one mapping

between $(X_t)_{1\leq t\leq T}$ and $(Y_t)_{1\leq t\leq T}$. In what follows we discuss the influence of the choice of the points $(\omega_1,\cdots,\omega_L)$ at which the Fourier transform is sampled.

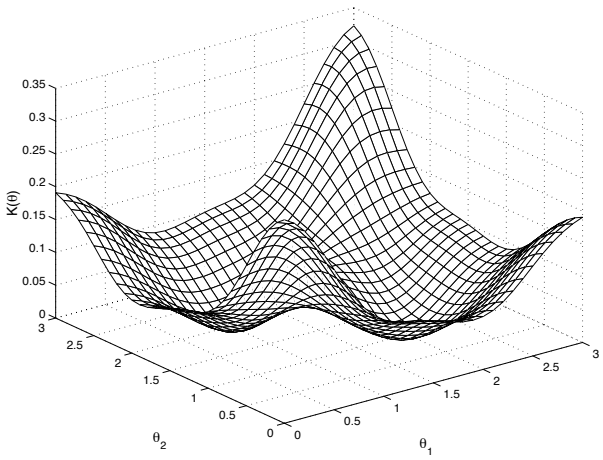


Figure 2: $\omega = 2\pi/X_{\max}, K = 2,\pi_1 = \pi_2 = 0.5,\lambda_1 = \lambda_2 = 1,\theta_1 = 1,\theta_2 = 2,T = 5000$

Note that $(\omega_1,\cdots,\omega_L) \rightarrow \alpha(\omega_1,\cdots,\omega_L)$results in a contraction of $\theta \rightarrow K(\theta)$ when $\alpha > 1$ (Fig. 2) and in a dilatation of $\theta \rightarrow K(\theta)$ when $\alpha < 1$ (Fig. 3). For the sake of simplicity we suppose in what follows that the sampling is regular $\omega_k = k\omega_1$. Let us study the effects of some variations on ω_1. First of all there are clearly more local extrema in $\theta \rightarrow K(\theta)$ as ω_1 increases because of the contraction effect mentioned above. The optimization algorithms consequently get trapped in the local extrema if ω_1 is too large. For example let us choose $T = 1000, L = 1, K = 2, \pi_1 = \pi_2 = 0.5, \lambda_1 = \lambda_2 = 2, \theta_1 = 1, \theta_2 = 2$. The initialization of θ is random with uniform distribution on $[0,3]\times[0,3]$. If the condition $\omega_1 < \frac{2\pi}{X_{\max}}$ is not fulfilled, the estimate $\hat{\theta}$ gets trapped in the local extrema of $K(\theta)$: for $\omega_1 = \frac{2\pi}{X_{\max}}\ \sigma_\theta^2 \simeq 10^{-3}$ and for $\omega_1 = 10\frac{2\pi}{X_{\max}}\ \sigma_\theta^2 \simeq 1$ where σ_θ^2 denotes the sum of the variances of the estimate of the offsets. It results from this discussion that ω_1 should be chosen as small as possible in order to avoid local extrema. But another effect of the choice of frequencies at which the Fourier transform is sampled is that the covariance matrix Γ has too high a condition number for optimization to be possible when ω_1 is very small. This point is discussed in what follows.

Condition number of the covariance matrix

Recall first of all that if the $(X_t)_t$ are i.i.d. then the matrix Γ has entry (i,j) $\Gamma(i,j) = \Phi(\omega_i - \omega_j)$. Since $\omega_k = k\omega_1$ the covariance matrix Γ tends to the rank one matrix ee^T where $e = [1,1,\cdots,1]^T$ when ω_1 tends to zero. The condition number of Γ is thus very high if ω_1 is too small. Table 1 provides the condition number of Γ for different values of $\omega_1 = \alpha 2\pi/X_{\max}$ and a growing number L of sampling points.

As one can see in Table 1 the condition number of Γ grows very quickly with the number L of frequencies at which the Fourier transform is sampled. A contradictory effect of L is that the variance of the estimate $\hat{\theta}$ is lowest if the number L of sampling points is high; the variance of the estimate decreases as $O(1/L)$ until $L = T$ and then the variance stabilizes. Table 2 shows this result.

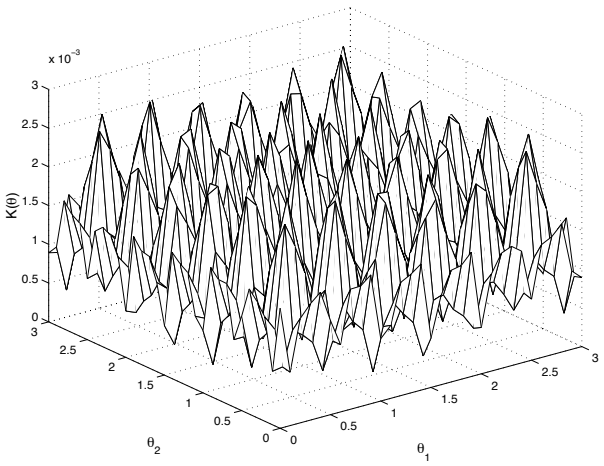


Figure 3: $\omega = 20 \times 2\pi/X_{\max}, K = 2,\pi_1 = \pi_2 = 0.5,\lambda_1 = \lambda_2 = 1,\theta_1 = 1,\theta_2 = 2,T = 5000$

L α	0.001	0.01	0.1	1	10	100	1000
2	10^5	10^4	100	1	1	1	1
5	10^{13}	10^9	10^5	10	1	1	1
10	10^{18}	10^{17}	10^9	100	10	1	1

Table 1: Condition number of Γ with $T = 1000; K = 2; \pi(1) = \pi(2) = 0.5; \lambda_1 = \lambda_2 = 2; \theta_1 = 1; \theta_2 = 2$.

In order to deal with the contradictory effects of the choice of the frequencies at which the Fourier transform is sampled we suggest the following approach. We successively estimate θ for simultaneously increasing values of α and L so that the condition number of $\hat{\Gamma}$ remains reasonable and we use the previous estimate of θ as an initial guess while changing α and L. The use of this previous estimate avoids converging to a local minimum of $K(\theta)$ when α increases. A possible alternative approach consists in replacing the consistent estimate $\hat{\Gamma}$ with the identity matrix in the criterion $K(\theta)$.

4.4. Least mean square estimation

In this Section we propose to undertake a least mean square estimation i.e. to minimize $M(\theta) = ||Z_T - m(\theta)||^2$.

4.4.1. Expectation Maximization method

The above problem is equivalent to maximizing the likelihood of $\hat{Z}_T$ w.r.t. θ for the following distribution :

$$\left\{\begin{array}{lcl} Z & = & W_1 + W_2 + \cdots + W_K \\ (W_k)_k & & independent \\ W_k & \sim & \mathcal{N}(m_k(\theta_k), \Gamma_k) \end{array}\right.$$

where $m_k(\theta_k) = \pi_k[e^{j\omega_1\theta_k}\Phi_{0,\lambda_k}(\omega_1),\cdots,e^{j\omega_L\theta_k}\Phi_{0,\lambda_k}(\omega_L)]^T$. An EM algorithm makes it possible to split the optimization of the likelihood into simpler one dimensional optimizations.

1. Expectation step

Compute $Q(\theta,\theta_k) = \mathbb{E}(L(W,Z;\theta) \mid Z=z;\theta_k) = C - \frac{1}{2}\sum_i \Gamma_i^{-1/2}\|m_i(\theta_i^k)-m_i(\theta_i)+\Gamma_i\Gamma^{-1}(z-\sum_{j=1}^K m_j(\theta_j^k)\|^2\Gamma_i^{-H/2}$.

2. Maximization step

If one supposes that the $W_i s$ all have the same covariance matrix $\Gamma_i = K^{-1}\Gamma$ the criterion to minimize w.r.t. θ_i reduces to $\|m_i(\theta_i^k) - m_i(\theta_i)) + K^{-1}(z - \sum_{j=1}^K m_j(\theta_j^k)\|^2$ and $\theta_i^{k+1} = Arg\min_{\theta_i}\|m_i(\theta_i^k) - m_i(\theta_i)) + K^{-1}(z - \sum_{j=1}^K m_j(\theta_j^k)\|^2$.

4.4.2. Simplified maximization step

The criterion $\|m_i(\theta_i^k) - m_i(\theta_i)) + K^{-1}(z - \sum_{j=1}^K m_j(\theta_j^k)\|^2$ can be reduced to a quadratic form in θ_i though yielding analytical reestimation for θ_i.

Denote by $\alpha_i^l = m_i(\theta_i) - K^{-1}(z - m(\theta^k))$ and by $\psi_l = Arg(\alpha_i^l)$ then $\alpha_i^l \simeq \pi_i \left|\Phi_{0,\lambda_i}(\omega_l)\right| e^{j\psi_l}$ and a second order development yields $\|m_i(\theta_i)-\alpha_i\|^2 \simeq \sum_l \pi_i^2 \left|\Phi_{0,\lambda_i}(\omega_l)\right|^2 (\omega_l\theta_i-\psi_l)^2$ and

$$\theta_i^{k+1} = \frac{\sum_l \left|\Phi_{0,\lambda_i}(\omega_l)\right|^2 \omega_l\psi_l}{\sum_l \left|\Phi_{0,\lambda_i}(\omega_l)\right|^2 \omega_l^2}$$

4.5. Example

The algorithm is used to estimate the offsets of a four component mixture of shifted exponential distributions. The parameters of offset are $\theta_1 = 1, \theta_2 = 2, \theta_3 = 3$ and $\theta_4 = 5$ and the intensities of the exponentials are all equal to $\lambda = 2$. The Fourier transform of the mixture is considered at only one ($L = 1$) pulsation $\omega = \frac{2\pi}{X_{\max}}$. The EM method 4.4.1 as well as the optimization of the simplified criterion 4.4.2 are considered. The Table below lists the mean and variance of the estimates of the offsets for $T = 1000$ independent realisations of the four component mixture. The performance of the algorithms in the case of small samples has also been investigated. For $T = 100$ independent realisations of the mixture the obtained bias is of the order of 0.1 times the true parameters of offset and the variance of the estimates is between 0.02 and 0.10 for both the EM and the simplified algorithm. The estimates at successive iterations of the EM algorithm are plotted in Figure 4 for small samples $T = 100$.

L	1	10	100	1000	10000	100000
σ_θ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-6}	10^{-6}

Table 2: Variance of $\hat{\theta}$. $T = 1000; K = 2; \pi(1) = \pi(2) = 0.5; \lambda_1 = \lambda_2 = 2; \theta_1 = 1; \theta_2 = 2; \omega = 2\pi/X_{\max}$

	Algorithm 4.4.1		Algorithm 4.4.2	
	$\mathbb{E}(\hat{\theta}_k)$	$var(\hat{\theta}_k)$	$\mathbb{E}(\hat{\theta}_k)$	$var(\hat{\theta}_k)$
θ_1	0.99	1.5×10^{-3}	1.06	1.9×10^{-3}
θ_2	2.21	1.7×10^{-3}	2.20	1.9×10^{-3}
θ_3	2.72	2.1×10^{-3}	2.70	2.5×10^{-3}
θ_4	5.06	6×10^{-3}	5.07	7.6×10^{-3}

Table 3: Mean and variance of the estimators.

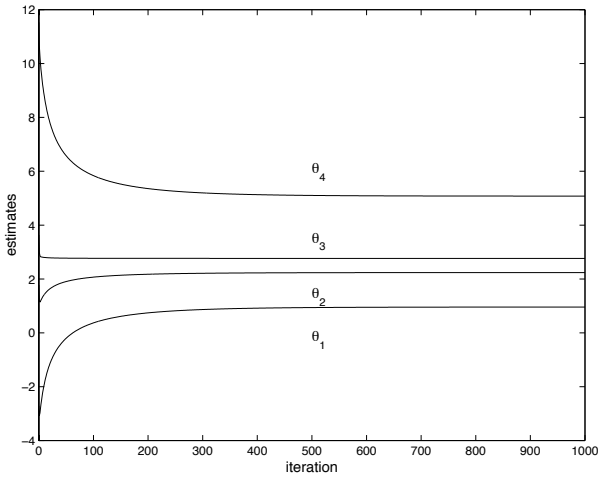


Figure 4: $T = 100$. $L = 1$. $\omega = 2\pi/X_{\max}$. $\theta_1 = 1, \theta_2 = 2, \theta_3 = 3, \theta_4 = 5$. $\hat{\theta_1} = 1.10, \hat{\theta_2} = 2.29, \hat{\theta_3} = 2.78, \hat{\theta_4} = 5.10$.

5. CONCLUSION

Time domain approaches do not enable the estimation of a mixture in some cases when offset parameters are unknown. We have proposed some Fourier transform based algorithms for estimating those parameters. When all but the offset parameters are known a good choice of sampling for the characteristic function results in no local extrema. In this context analytical reestimation formulae have been derived thus permiting estimation with extremely low computational burden and fast convergence.

6. REFERENCES

[1] J.L. Bryant and A.S. Paulson, "Some comments on characteristic-function based estimations," *Sankhya A*, vol. 41, pp. 109–116, 1979.

[2] S.Vaton, H.Korezlioglu, and T.Chonavel, "Modelling lan traffic data as a locally stationary semi-markov process," in *Performance Modelling and Evaluation of ATM Networks Vol.3*, D.D.Kouvatsos, Ed. 1997, Chapmann and Hall.

[3] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood estimation from incomplete data via the em algorithm," *JRSS-B*, vol. 39, pp. 1–38, 1977.

[4] G. Celeux and J. Diebolt, "The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem," *Comp. Stat. Quat.*, vol. 2, pp. 73–82, 1985.

[5] E.Pitman, "The estimation of location and scale parameters of a continuous population of any given form," *Biometrika*, , no. 30, pp. 391–421, 1939.

[6] T.L. Marzetta, "Computing the barankin bound by solving an unconstrained quadratic optimization problem," in *ICASSP'97*, Munich, Germany, april 1997, pp. 3829–3832.

[7] P. Doukhan, *Mixing: properties and examples*, Lecture Notes in Statistics. Springer-Verlag, 1994.

Hidden Markov Modeling for network communication channels

Kavé Salamatian
Laboratoire LIP6-CNRS UMR7606
Université Pierre et Marie Curie
8 rue du Capitaine Scott
75015 Paris, France.
Kave.Salamatian@lip6.fr

Sandrine Vaton
ENST Bretagne
BP 832
29285 Brest,France.
Sandrine.Vaton@enst-bretagne.fr

ABSTRACT

In this paper we perform the statistical analysis of an Internet communication channel. Our study is based on a Hidden Markov Model (HMM). The channel switches between different states; to each state corresponds the probability that a packet sent by the transmitter will be lost. The transition between the different states of the channel is governed by a Markov chain; this Markov chain is not observed directly, but the received packet flow provides some probabilistic information about the current state of the channel, as well as some information about the parameters of the model. In this paper we detail some useful algorithms for the estimation of the channel parameters, and for making inference about the state of the channel. We discuss the relevance of the Markov model of the channel; we also discuss how many states are required to pertinently model a real communication channel.

Keywords

Hidden Markov Model, Internet modelling, active measurement Expectation-Maximization, network state estimation.

1. INTRODUCTION

Recently, much research effort has been spent on the performance analysis of IP networks. These efforts resulted in the IPPM group of the IETF defining different end-to-end performance metrics [10]. However, these metrics are meaningless without any modeling to translate the measured metrics fluctuations into network states. The model can be seen here as an "Occam's razor", describing in a compressed and concise manner an IPPM metrics trace gathered on an Internet path.

This paper presents the EM (*Expectation-Maximization*) algorithm as a generic approach for inferring about unseen parameters based on observed IPPM metrics.

Another domain where modeling is extremely important is the field of adaptive multimedia network applications. Network heterogeneity is a fact of life in today's Internet. Indeed, the current Internet provides users with only a single class of best effort service which does not promise anything in terms of guaranteed performance. Measurements show persistent problems with multimedia quality caused by congestion in the network, and thus by the impact of traffic in the network on any application stream. This impact is felt through high loss rates, varying delay, *etc.* The de-centralized nature of the Internet makes it very likely that it will continue to be as unpredictable in the future.

The unpredictable nature of network fluctuations makes it difficult for applications to determine in advance the network performance between a pair of Internet hosts. So, in order to always provide the best possible quality, applications have evolved to adapt dynamically to their current network environment. The TCP congestion avoidance mechanism is a good example of such an adaptive mechanism which aims to adapt the rate of transmission to the state of the network.

Network state is not a concrete and well defined notion. In fact, network state is an *abstract* variable, representative of the effect of all concurrent flows on one application flow. As applications have no direct access to information on router loads and characteristics, the network state is a hidden variable, that can be perceived by an application only through its effects on its data flow. Packets can thus be viewed as probes that give incomplete and delayed information about the path they have crossed. Active IP performance measurement tools generate such probe traffic to estimate the overall performance of an Internet path. Adaptive applications have to evaluate the network conditions from the information they can gather from the received packet flows, and adapt to it.

For example, in TCP, the network state is a binary value (*congested* or *non congested*) and is estimated by monitoring packet losses: a single loss observed at time T is interpreted by the TCP congestion avoidance mechanism as a *congested* state for the network at time T. Even in such a case, the network state at time T is a hidden variable estimated by a delayed observation on a single packet loss.

Two kinds of information can be extracted from the received packet flow: packet loss process and packet delay. Due to non-synchronized clocks between receivers and senders, the reliable measurement of packet delay is difficult. Strict synchronization of two entities connected by a varying delay link, can prove to be impossible without access to an external universal time reference as provided by a GPS (*Global Positioning System*) time reference [2]. In [9], complex mechanisms that converge asymptotically to the synchronization of two clocks are developed. But GPS acquisition cards have not been widely deployed in the Internet, making delay measurements unreliable. Moreover, delay is a continuous variable, making state estimation based on it even more complex.

In this paper, we present an analysis of the end-to-end loss process and use it to make inferences about the state of the network as seen by the application.

Previous empirical measurement have shown that the distribution of the number of consecutive lost packets is approximately geometric, or, rather, that the head of the distribution is geometric, and that the tail includes a few events (which might contribute significantly to the overall loss rate, since a single event in the tail indicates a loss burst with a large number of lost packets) which appear not to have any specific structure [13, 3, 1, 15, 16]. Unfortunately, the above result says little about the characteristics of the loss process because it only applies to the marginal distribution of the process and says nothing about the correlation structure of that process.

Most of the previous works have tried to model the loss sequence by a non Hidden Markov Model. [16] proposes a classical non Hidden Markov model of the Internet channel with 2^k states, with k in the range $[0, 6]$. In [16], the observed loss value at time t, $Y(t)$ depends on $(Y(t-1), Y(t-2), ..., Y(t-k))$. This leads from a loss sequence of loss T to a Markov chain model with 2^k states and a finite memory (at most k packets) in the observed loss process that can be estimated with a complexity that is on the order of $\mathcal{O}(T)$. Unlike these models, HMMs (*Hidden Markov Models*) exhibit infinite dependences in the observed process, even with only 2 states. This strongly reduces the number of states needed to describe a given loss process. We will show further that in almost all cases studied fewer than 4 states are sufficient, where [16] has used up to 42 states. Moreover, a non hidden model cannot be used for state estimation and is thus only descriptive. This smaller number of state in HMMs should be traded off with the higher complexity induced by the hidden parameters estimation. The complexity of HMM estimation is on the order of$\mathcal{O}(K^2T)$, where K is the number of states of the HMM.

In this paper, we study more sophisticated models based on HMMs. HMMs have proved to be useful for a number of applications. One can cite different examples of their application in the field of telecommunications, including speech recognition [11], traffic characterization [14], source coding, channel coding, and equalization.

We will develop a two step network state estimation procedure: first, a model calibration step that chooses a number of states and calibrates an HMM for the loss, and a second step that will use this HMM to estimate the actual (or past) state of the network by observing the sequence of lost packets.

The paper is organized as follows. We will first introduce our underlying model and HMMs. After that, we will define an estimator for the number of states of an HMM. In section 4, we will describe the EM (*Expectation-Maximization*) algorithm in the context of the calibration of HMMs for network channel modeling. Next, we will investigate state sequence inference using observed lost packet sequences. We will continue by illustrating the concepts developed in the paper with real traces collected from the Internet. A final discussion and some conclusions are presented in section 7.

2. MODELING OF NETWORK CHANNELS

A clear explanation of the underlying network model as seen by the application is essential in order to remove ambiguities. In this work, the network is modelled as a valve that can be passing or blocking at any time. The state (*passing* or *blocking*) of the valve at a time t is $S(t)$. The packet flow generated by our application generates packet i at time T_i and samples the state of the valve $S(T_i')$ at time T_i', where $T_i' - T_i$ represents the delay needed to reach the bottleneck, where packets are lost. We will suppose in what follows that the sampling time is slided back to the sending time $T_i' = T_i$. Indeed, a more complete analysis require knowledge of the distribution for the sampling delay $T_i' - T_i$.

The state estimation process tries to estimate statistical characteristics of the open/close process governing the valve, based on the observed packet loss process, which is the sampled process $S(T_i)$. We make the essential hypothesis that the open/close process has reached a stable and stationary distribution. This stationary distribution is, of course, a function of the competing Internet traffic and of the traffic generated by our application. However, active probing IP performance measurement tools try to maintain low traffic to avoid disturbing the open/close stationary distribution.

In the general case, we suppose that a loss trace contains K samples $\{S(T_i)\}$, $i = 1, \ldots, K$. Under ergodic and stationary hypotheses for the open/close stochastic process, it is possible to estimate the statistical characteristic of $S(t)$, based on the samples. For example, the temporal mean $\bar{S} = \frac{1}{K}\sum_{i=1}^{K} S(T_i)$ is an unbiased estimator of the open/close process mean $\mu = \mathrm{E}\{S(T)\}$. The variance of this estimator will be $\mathrm{var}\{\bar{S}\} = \frac{1}{K^2}\sum_{i,j=1}^{K} R(|T_i - T_j|)$, where $R(\tau) = \mathrm{E}\{(S(T+\tau) - \mu)(S(T) - \mu)\}$ is the autocorrelation of the open/close process.

Actually, active probing techniques attempt to extend inferences made on the loss process seen by the probing flow, to other competing flows. The previous remark helps determine conditions for this extension. All competing flows are governed by the same open/close process, therefore under the stationary hypothesis for the open/close process, the temporal mean estimator of all flows will indeed converge to the same value. However the variance of this estimator will largely depend on the autocorrelation function of

the open/close process and on the dynamics of the particuliar flow. For example, TCP flows that send a bunch of very close packets on the window opening will undergo higher estimate variance than competing UDP flows that send packets more regularly. This remark does not mean that a UDP flow will see lower loss rates than a TCP flow, it only says that if a UDP and a TCP flow are competing, the TCP flow may see a larger fluctuation of its loss rate than the more regularly spaced UDP flow.

If delay information is available, one can extend the above model to take it into account. In this case, the network can be modeled as an valve which is governed by an open/close process, with a delay element. The state estimation procedure will try to estimate the joint open/close and delay process using the observed measures. However, recent empirical studies have shown that delay and loss rates are statistically independent [9, 1]. This is mainly due to the fact that losses and delays do not occur at the same location in the network. Based on this empirical observation, the joint estimation problem would be slipt into two independent estimation problems, one for the loss process and the other for the delay. This separation procedure is in fact used in the TCP context : congestion detection which is based on the loss process is independent of delay estimation which is based on *rtt* exponential smoothing. In this paper we only consider the loss process modeling and estimation problem. The delay modeling can use the same approach, but it is more complex as the delay is a continuous variable.

We suppose in what follows that the open/close process in our model of the network follows a Markov chain. This is a very weak hypothesis, as a Markov chain of arbitrarily large order can be used to model an extremely wide range of processes [5]. We consider an HMM for the channel between a transmitter and a receiver on the Internet. The intuition behind our approach is that at regular sampling times, the open/close process will pass through different (hidden) states which are reflected in the observable loss rate fluctuations.

The loss process $\mathbf{X} = (X_t)_{t=1}^{T}$ is defined as $X_t = 0$, if the t^{th} packet reaches its destination and $X_t = 1$, if the packet is lost. The channel '*switches*' between K different states following a Markov chain $\mathbf{Y} = \{Y_t\}$ with state space $\mathcal{S} = \{1, 2, \cdots, K\}$ and stochastic transition matrix $\mathbf{\Gamma}(t) = (\Gamma_{ij})_{i,j=1}^{K}$ where $\Gamma_{ij} = \text{Prob}\{Y_{t+1} = j \mid Y_t = i\}$. The Markov chain is homogeneous, ergodic and the state distribution converges to a stationary distribution π which is the solution of the following set of equations : $\pi.\mathbf{\Gamma} = \pi$ and $\pi.[1, 1, \cdots, 1]^T = 1$.

In each of these K states, the channel is uniformly blocking or passing. This defines a probability that a packet is lost at time while the channel is in state i ($1 \leq i \leq K$); these probabilities are grouped in the observation matrix $\mathbf{P} = (p_i)_{i=1}^{K}$ where $p_i = \text{Prob}\{X_t = 1 \mid Y_t = i\}$. In what follows we will represent a sub-vector $(x_i, x_{i+1}, \cdots x_j)$, $i < j$ by $\mathbf{x}_i^j$.

To model the network channel by a HMM, we need a procedure for estimating the number of states (K), the transition matrix ($\mathbf{\Gamma}$) and the observation matrix ($\mathbf{P}$) of the Markov chain. Assuming that the number of states is K, we denote the set of parameters by $\theta = (\mathbf{\Gamma}, \mathbf{P})$. We will study these estimation problems in the following sections.

3. ESTIMATION OF THE NUMBER OF STATES OF A HIDDEN MARKOV MODEL

In order to choose a correct number of state for the HMM, we need a consistent estimator, based on the empirically observed loss process $\mathbf{X}$. In this paper, we use an estimator developed in [18] which is based on the notion of entropy and data compression. The entropy of a discrete random variable X is defined as :

$$H(X) = -\sum_{x} \text{Prob}\{X = x\} \log \text{Prob}\{X = x\}$$

The entropy of a stationary stochastic process $\mathbf{X} = \{X_i\}$ is defined as :

$$H(\mathbf{X}) = \lim_{n \to \infty} \frac{1}{n} H(X_1, \cdots, X_n)$$

where $H(X_1, \cdots, X_n)$ is the entropy of the joint random variable $(X_1, \cdots, X_n)$.

We know from the AEP (*Asymptotic Equipartition Property*) [4] that if the stochastic process $\mathbf{X}$ is finite-valued, stationary and ergodic then :

$$-\frac{1}{n} \log \text{Prob}\{\mathbf{X} = \mathbf{x}\} \to H(\mathbf{X}) \text{ with probability 1}$$

On the other hand, the compression theorem states that the optimal code length $l_n^*(\mathbf{x})$ of a compression scheme converges toward the entropy of the stochastic process $\mathbf{X}$:

$$l_n^* \to H(\mathbf{X})$$

Based on the preceding preliminaries we define the following estimator for the number of states K^* of a HMM :

$$K^* = \text{Arg min} \left\{ j \Big| -\frac{1}{n} \log \max_{\theta} \text{Prob}_{j,\theta}\{\mathbf{X} = \mathbf{x}\} - H(\mathbf{X}) \Big| < \epsilon_n \right\} \quad (1)$$

where $\max_\theta \text{Prob}_{j,\theta}\{\mathbf{X} = \mathbf{x}\}$ is the probability of observing the output sequence $\mathbf{x}$ for an HMM with j states under the parameter θ that maximizes the likelihood of observing the output sequence $\mathbf{x}$. ϵ_n is an arbitrary sequence converging towards 0.

The intuition behind this estimator is that a good guess for the number of states will results in an optimal code-length (l_n^*), close to the entropy of the stochastic process.

This estimator has been shown in [18] to be consistent and asymptotically optimal.

Two main remaining problems about the above defined estimator are how to choose the parameter θ that maximizes the likelihood of the observed sequence $\mathbf{x}$ and how to estimate the entropy of the stochastic process.

The first problem will be addressed in the following sections. The second problem is resolved using a universal compression encoding such as Lempel-Ziv [17] or an arithmetic coding [8]. The main idea is that if the observed sequence $\mathbf{x}$ is sufficiently long then the mean code-length of an encoding obtained by a universal compression scheme

will converge toward the entropy of the stochastic process. Using this observation and assuming that we have sufficient data to make the mean code-length converge, the estimator is rewritten as :

$$K^* = \text{Arg}\min \{j \mid -\frac{1}{n}\log \max_{\theta} \text{Prob}_{j,\theta}\{\mathbf{X}=\mathbf{x}\} - \frac{1}{n} l^n(\mathbf{x}) \mid < \epsilon_n \} \quad (2)$$

where $l^n(\mathbf{x})$ represents the length of the encoding of the output sequence $\mathbf{x}$.

However the convergence speed of the mean code length can be slow, and a long sequence should be used to attain a stable estimated entropy. Classical results in the theory of Lempel-Ziv coding show that the estimation error made by replacing entropy by the mean code-length of a Lempel-Ziv encoding is of order $\mathcal{O}(\frac{\log\log n}{\log n})$ [17]. However this error is of order $\mathcal{O}(\frac{\log n}{n})$ for arithmetic coding [8].

Another estimator of the number of states of an HMM is proposed in [6]. In contrast to the estimator in Eq. 1 which needs an estimate of the likelihood, the estimator proposed in [6] does not. However, in our case, we already obtain this estimate as a byproduct of the estimation of the HMM parameters. This estimation will be described in the following section.

4. INFERENCE ABOUT THE PARAMETERS OF THE CHANNEL: THE EM ALGORITHM

As the states of the Markov chain are not observable we must estimate the set of parameters θ using the information hidden in the loss process. The estimation criterion is the Maximum Likelihood criterion. This criterion chooses the parameters $\hat{\theta}$ that maximize the probability of seeing the observed loss process $\mathbf{X}$ given that the HMM follows these parameters. The estimation procedure then reduces to an optimization problem with a simple cost function.

The Expectation Maximization (EM) algorithm [11] is a valuable approach for maximum likelihood parameter estimation in mixture models and in various type of Markov modulated models Markov Modulated Poisson Processes, Markov Arrival Processes, *etc*).

It has a number of advantages, especially its stability : each iteration of the EM algorithm increases the likelihood of the model; this ensures the convergence of the algorithm to a local, but not necessarily global extremum of the likelihood. Another major advantage of the EM algorithm is its numerical complexity. A direct computation of the likelihood would require K^T terms, where K is the number of different states for the underlying process, and T is the number of observations. On the other hand, the numerical complexity of the EM algorithm is of the order of K^2T.

The EM algorithm involves maximizing iteratively with respect to θ a function $Q(\theta,\hat{\theta}_k) = \text{E}\{L(\mathbf{X},\mathbf{Y};\theta) \mid \mathbf{X}=\mathbf{x};\hat{\theta}_k\}$. In this expression $\mathbf{Y}$ is the unobserved Markov state sequence, $\mathbf{X}$ is the vector of observations -a probabilistic function of $\mathbf{Y}$- and $L(\mathbf{X},\mathbf{Y};\theta)$ denotes the log-likelihood of the 'complete data' $\mathbf{Z}=(\mathbf{X},\mathbf{Y})$ when θ is the parameter of the model. Expectation involved in the computation of $Q(\theta,\hat{\theta}_k)$ is the expectation given that $\mathbf{X}=\mathbf{x}$ and given that $\hat{\theta}_k$ is the parameter of the model.

The dependence structure of the HMM (see Fig. 1) is such that the complete log-likelihood $L(\mathbf{X},\mathbf{Y};\theta)$ can be split into two terms $L(\mathbf{X},\mathbf{Y};\theta) = L(\mathbf{Y};\theta) + L(\mathbf{X}\mid\mathbf{Y};\theta)$.

Each iteration of the EM algorithm can consequently be decomposed into two steps :

1. Step E (Expectation):
   Compute $Q(\theta,\hat{\theta}_k) = \text{E}\{L(\mathbf{X},\mathbf{Y};\theta) \mid \mathbf{X}=\mathbf{x},\hat{\theta}_k\}$.
2. Step M (Maximization) :
   Maximize $Q(\theta,\hat{\theta}_k)$ with respect to θ :
   $$\hat{\theta}_{k+1} = \text{Arg}\max_{\theta} Q(\theta,\hat{\theta}_k).$$

The maximization involved in the M step is analytical and does not require intensive computation; the integration involved in the E step requires the computation of a non linear filter; this computation is based on the Forward Backward (or Baum-Welches) algorithm [11].

The application of the general EM framework to the estimation the HMM model for the loss process is non trivial and awkward. For the sake of completeness, we present the full derivation. To simplify the notation, we use the following convention :

$$1\{x,p\} \triangleq 1\{x=1\}p + 1\{x=0\}(1-p)$$
$$\log 1\{x,p\} \triangleq 1\{x=1\}\log p + 1\{x=0\}\log(1-p)$$

Denote by $\psi_t^k(i,j)$ and by $\gamma_t^k(i)$ the *a posteriori* probabilities given the parameter $\hat{\theta}_k$ and the observation $\mathbf{x}$:

$$\psi_t^k(i,j) = \text{E}\{1\{Y_t=i, Y_{t+1}=j\} \mid \mathbf{X}=\mathbf{x};\hat{\theta}_k\}$$
$$\gamma_t^k(i) = \text{E}\{1\{Y_t=i\} \mid \mathbf{X}=\mathbf{x};\hat{\theta}_k\}.$$

With these notations we have :

$$L(\mathbf{Y};\theta) = \sum_{i,j=1}^{K} \log\Gamma_{ij} \sum_{t=1}^{T} \psi_t^k(i,j),$$
$$L(\mathbf{X}\mid\mathbf{Y};\theta) = \sum_{t=1}^{T} \gamma_t^k(i) \log 1\{X_t, p_i\}.$$

Maximizing $L(\mathbf{Y};\theta)$ with the constraint that $\sum_{j=1}^{K}\Gamma_{ij}=1$ results in a new estimate of $\hat{\mathbf{\Gamma}}(k+1)$ such that :

$$\hat{\Gamma}_{ij}(k+1) = \frac{\sum_{t=1}^{T}\psi_t^k(i,j)}{\sum_{t=1}^{T}\sum_{j=1}^{K}\psi_t^k(i,j)}. \quad (3)$$

Maximizing $L(\mathbf{X}\mid\mathbf{Y};\theta)$ results in a new estimate of $\hat{\mathbf{P}}(k+1)$ such that :

$$\hat{p}_{k+1}(i) = \frac{\sum_{t=1}^{T}\gamma_t^k(i)1\{X_t=1\}}{\sum_{t=1}^{T}\gamma_t^k(i)}. \quad (4)$$

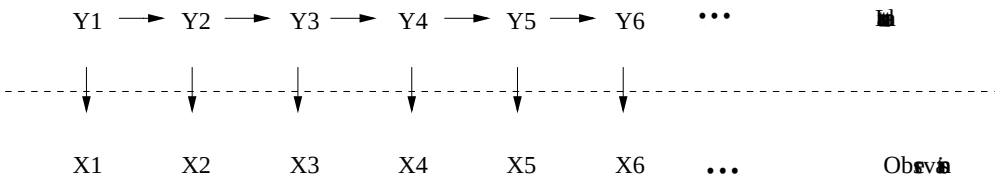


Figure 1: Dependence structure of the Hidden Markov Model.

The computation of the *a posteriori* probabilities $\psi_t(i,j)$ are performed by means of a Forward-Backward algorithm.

Let us introduce the 'forward' filter

$$\alpha_t^k(i) = \text{Prob}\{\mathbf{X}_1^t, Y_t = i \mid \hat{\theta}_k\}$$

and the 'backward' filter

$$\beta_t^k(i) = \text{Prob}\{\mathbf{X}_{t+1}^T \mid Y_t = i,\ \hat{\theta}_k\}$$

. It results from the weak Markov property that α and β can be computed recursively :

$$\alpha_{t+1}^k(i) = 1\{X_{t+1}, \hat{p}_i(k)\} \sum_{j=1}^{K} \alpha_t(j)\hat{\Gamma}_{ji}(k) \tag{5}$$

$$\beta_t^k(i) = \sum_{j=1}^{K} \hat{\Gamma}_{ij}(k)\beta_{t+1}^k(j)1\{X_{t+1}, \hat{p}_j(k)\} \tag{6}$$

$\alpha(.)$ is computed in the 'forward' direction

$$\alpha_{t+1}^k(.) = F(\alpha_t^k(.), X_{t+1}, \hat{\theta}_k)$$

and $\beta(.)$ is computed in the 'backward' direction

$$\beta_t^k(.) = G(\beta_{t+1}^k(.), X_{t+1}, \hat{\theta}_k)$$

. ψ and γ can be deduced straightforwardly from the α and β filter as follows :

$$\psi_t^k(i,j) \propto \alpha_t^k(i)\hat{\Gamma}_{ij}(k)\beta_{t+1}^k(j)1\{X_{t+1}, \hat{p}_j(k)\} \tag{7}$$

$$\gamma_t^k(i) \propto \alpha_t^k(i)\beta_t^k(i) \tag{8}$$

where $\propto$ denotes the equality up to a multiplicative factor such that $\sum_{i,j=1}^{K} \psi_t^k(i,j) = 1$ and that $\sum_{i=1}^{K} \gamma_t^k(i) = 1$. Fig. 2 illustrates the successive steps of the HMM parameter estimation process.

The next step involves inference about the state of the channel at each step based on the observed output sequence $\mathbf{x}$ and an *a priori* HMM for the channel.

5. INFERENCE ABOUT THE STATES OF THE CHANNEL

In this section we will investigate the online inference about the (hidden) state of network channel. We assume here that the channel is stationary and that we have already calibrated an HMM for it using an avaiable observed sequence. We want to make inference about the channel state only on the basis of the forthcoming loss process. Two approaches will be described in this section: the first one, the Marginal Posterior Mode MPM, attempts to estimate the most probable state at step t ($\hat{\mathbf{Y}}_t$) based on the observed loss process up to step t ($\mathbf{x}_0^t$). The second approach, based

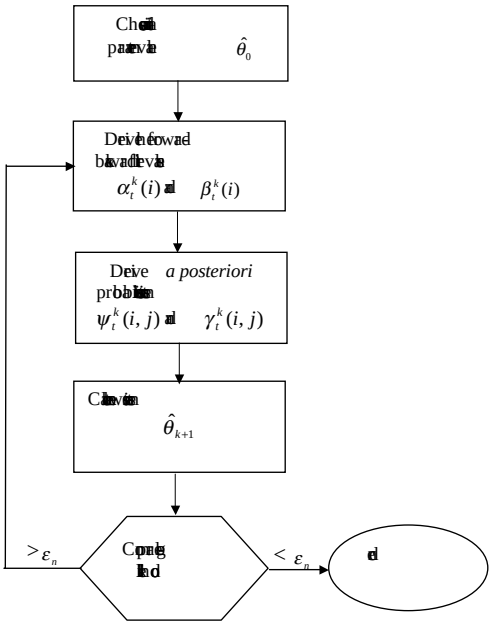


Figure 2: Summary of the estimation of Hidden Markov Model parameters

on the Viterbi algorithm, estimates the most probable state sequence $\hat{\mathbf{Y}}_0^T$ using the observed loss process $\mathbf{x}_0^T$.

5.1 The Marginal Posterior Mode

The Forward Backward algorithm produces as a byproduct the *a posteriori* Marginal distribution $\gamma_t(i) = \text{Prob}\{Y_t = i \mid \mathbf{X}\}$ which leads to the MPM estimate which is the most probable state at time t, given the observed sequence $\mathbf{X}$. This estimate is the maximizer in i of $\gamma_t(i)$:

$$\hat{s}_t^{MPM} = \text{Arg} \max_{1 \le i \le K} \gamma_t(i). \tag{9}$$

What is more, $\gamma_t(\hat{s}_t^{MPM})$ is a measure of the reliability of this estimate.

The state estimatation at time t by MPM is based on the forward-backward filter (Eq. 8). The effect of the past observations are summarized in $\alpha_t(i)$, $i = 1, \ldots, K$ and a look-ahead window of size W is used to derive $\beta_t(i)$. The look-ahead window width can be as low as 0; in this case $\gamma_t(i) = \alpha_t(i)$. However more robust estimation will need a look-ahead window.

The recursive structure of the forward backward filter greatly

reduces the complexity of updating α, β and γ.

The MPM makes possible online *a posteriori* estimation of the state of the network channel (with a look ahead window of size 0). This estimate can be used for adaptation purposes by an adaptive application.

For example, $\hat{s}_t^{MPM}$ can be viewed as a congestion indicator which is more reliable than a single packet loss, as used in TCP.

5.2 The Viterbi algorithm

Another approach to state estimation is based on the Viterbi algorithm. This algorithm produces a sequence

$$\hat{\mathbf{Y}} = (\hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_T)$$

that is globally the most likely given $\mathbf{X} = \mathbf{x}_0^T$. This estimate produced by the Viterbi algorithm will in general not coincide with the Marginal Posterior Mode at any time index t. The sequence produced by the Viterbi algorithm is the maximizer of the *a posteriori* log-likelihood $L(\mathbf{Y} \mid \mathbf{X} = \mathbf{x}; \theta)$ whereas the Marginal Posterior Mode for time index t is the maximizer of $L(Y_t \mid \mathbf{X} = \mathbf{x}; \theta)$. In general, the sequence produced by the Viterbi algorithm presents longer homogeneous intervals than the sequence produced by the Maximum Posterior Mode criterion; but the Maximum Posterior Mode criterion is less sensitive to inadequacy of the data to the HMM, in particular when the sojourn times of the underlying process are not geometric as is imposed by Markov models, but have another distribution (Pareto or any other heavy-tailed distribution). In the latter case, Hidden Semi-Markov Models would be more appropriate than the simpler HMMs.

The *a posteriori* log-likelihood $L(\mathbf{Y} \mid \mathbf{X} = \mathbf{x}; \theta)$ is equal to the complete log-likelihood up to an additive constant, $L(\mathbf{Y} \mid \mathbf{X} = x; \theta) = L(\mathbf{Y}, \mathbf{X} = \mathbf{x}; \theta) - L(\mathbf{X} = \mathbf{x}; \theta)$ so that the maximizer of $L(\mathbf{Y} \mid \mathbf{X} = \mathbf{x}; \theta)$ is the maximizer of $L(\mathbf{Y}, \mathbf{X} = \mathbf{x}; \theta)$. The dependence structure of HMM (Fig. 1) leads to an additive expression for the complete log-likelihood:

$$L(\mathbf{Y}, \mathbf{X} = \mathbf{x}; \theta) = \sum_{t=1}^{T} \big(\log \mathrm{Prob}\{Y_{t+1} \mid Y_t\} + \log \mathrm{Prob}\{x_{t+1} \mid Y_{t+1}\} \big).$$

This sum can be represented graphically as the length of a path in a lattice whose form is displayed in Figure 3. In this trellis, the node (t, i) corresponds to the state $Y_t = i$ of the channel; the branch that goes from the node (t, i) to the node $(t+1, j)$ corresponds to the transition $Y_t = i$ to $Y_{t+1} = j$. With this branch is associated a length such that:

$$d\big((t, i) \to (t+1, j)\big) = \log \Gamma(i, j) + \log 1\{x_t, p_j\}, \qquad (10)$$

The complete log-likelihood $L(\mathbf{Y}, \mathbf{x}; \theta)$ is the total length of the path $\mathbf{Y}$ in the lattice. The Viterbi algorithm retrieves the longest path in this lattice.

The additive form of the criterion makes it possible to construct a dynamic programming algorithm to solve this optimization problem [12].

The estimate returned by the Viterbi algorithm can also be used as a network state estimate in adaptive applications. The reception of each new packet leads to a recomputation of the longest path in the lattice. Nevertheless, in most cases only the last steps of the path change and one can restrict the path recomputation to a look-back window of fixed size. This means that the Viterbi algorithm induces an estimation latency that is incompatible with online estimation as an M packet look-ahead window induces a delay of $M\Delta$ (where Δ is the interpacket delay) on a stable state estimation. Nevertheless, the complexity induced by the Viterbi algorithm remains small.

6. SIMULATION RESULTS

6.1 Estimation of simulated packet loss traces

In this section we will put into practice concepts developed in previous sections. To validate our approach we have applied our estimation procedure over a simulated packet loss trace with a known HMM parameter. For this purpose, we generated a packet loss trace following a 3-state Hidden Markov Model with parameters shown in Table 1 (π is the stationary distribution of the Markov chain). We have then applied the EM based HMM model estimation to derive the estimated HMM parameters that are compared with the initial parameters values. Using an arithmetic compression

$$\mathbf{\Gamma} = \begin{bmatrix} 0.9 & 0.08 & 0.02 \\ 0.3 & 0.6 & 0.1 \\ 0.25 & 0.45 & 0.3 \end{bmatrix}$$

$$\mathbf{P} = (0.05, 0.20, 0.95) \quad \pi = (0.744, 0.206, 0.050)$$

Table 1: Parameters of simulated HMM

code, the entropy of the simulated loss trace was estimated to be $\hat{H} = 0.5430$. The EM algorithm was then applied to a sequence of length 10000 generated following the above HMM, for a number of states up to 4. The results (observation vector estimate $\hat{\mathbf{P}}$ and stationary distribution $\hat{\pi}$) are summarized in Table 2. It can be seen from this table that the log-likelihood of the estimated HMM converges to the entropy of the packet loss trace when the number of states increases. The 3-state HMM estimate is very close to the initial simulated HMM. However, as a 4 (or more) state model includes a model with fewer states, the quality of the estimate (measured by the log-likelihood) is better for 4 than for 3 states. But this should be traded-off against the increased complexity of a higher state model. The relative benefit of adding one state and passing from 3 to 4 is only around 0.01 %. But the relative benefit of going from 2 to 3 states is 0.3 %. We can then safely choose a 3-state HMM estimate for the simulated packet loss trace. The parameter ϵ_n in (Eq. 1) captures the trade-off between higher accuracy and higher complexity.

The EM algorithm has a major limitation: it is generally sensitive to its initialization. We checked for this particular data, that convergence does not depend on initialization. Our statistical interpretation is that the likelihood does not have many local extrema. This means that the different states (blocking/passing/intermediate...) can be well identified and that the Markov model is reasonable for the data we consider.

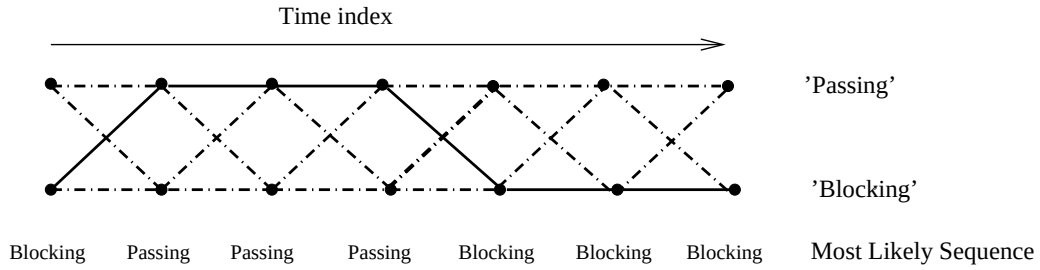


Figure 3: Viterbi algorithm for a two state model (the longest path is in the solid line).

Number of states	Log-Likelihood	Vector $\hat{\mathbf{P}}$	State distribution $\hat{\pi}$
1	0.5514	(0.1278)	(1)
2	0.5452	(0.067, 0.7940)	(0.91, 0.09)
3	0.5435	(0.03, 0.22, 0.97)	(0.74, 0.20, 0.06)
4	0.5434	(0.05, 0.20, 0.66, 0.93)	(0.80, 0.11, 0.06, 0.03)

Table 2: Results of the EM algorithm applied to simulated packet loss traces for up to 4 states

We also use the simulated trace to validate our state estimation procedure described in Section 5. For this purpose the state transitions estimated by MPM (without any look-ahead windows) and the Viterbi algorithm (with a look-ahead window of 10 time slots) are compared with the real simulated state transition. This comparison is shown in Fig. 4. As underlined in section 5.2, the transition sequence estimated by the Viterbi algorithm is more homogeneous than the one estimated in MPM, but the MPM sequence is more dynamic and follows more faithfully the original state transition. As can be expected, transitions with a short sojourn time are not easily detected by the estimation procedure, as they are too volatile to be detected. Another interesting feature is that, almost all of the time, transitions are detected fast (with no more than a 1 or 2 time slot delay). This is compatible with the application of the state estimation procedure in an adaptive application. However, for the Viterbi estimator, a look-ahead window of 10 time slots is needed to stabilize the estimation.

6.2 Estimation of real packet loss traces

We also used loss traces collected from the Internet following the IPPM Metrics recommendation [10]. We sent over the Internet a sequence of regularly spaced packets of equal size from Paris to different addresses in the United States and Europe. The packets were regularly spaced with a delay between packets of $\Delta = 40, 50, 100$ msec. We analyzed the sequence of zeros and ones that correspond to the reception or non reception of packets.

The methodology described in this paper was applied to a sequence of 10000 packets (corresponding to a period of 500 seconds with a constant inter packet time of 50 msec). We describe here two applications of the procedure to two traces. These traces are described in Table 3. The EM iterative process is implemented using Matlab [7] and applied to these traces with a stopping criterion of relative likelihood gain less than 0.01%. This leads to fewer than 20 iterations in most cases and less than one minute calculation for the calibration of a 4 state HMM. It is clear that a C language implementation of the estimation procedure will greatly speed up this time. We never had to use more than 4 states in the 36 traces we analyzed.

Trace	Interval ($msec$)	Mean loss rate	Estimated Entropy
1	50	14%	0.51048
2	50	18.83%	0.6150

Table 3: Basic parameters of 2 selected traces

Table 4 displays the parameters estimated by the EM algorithm for trace 1. We performed the estimation in a model with $K = 2$ states, with $K = 3$ states and with $K = 4$ states.

First, the EM is trained under the assumption that the channel switches between $K = 2$ states (blocking and passing). The probability of loss in the blocking state is estimated to be 0.9466 whereas the probability of loss in the second state is 0.110. It blocks 1.9% of the time. The average sojourn time in the blocking state is equal to 21 time slots (1050 msec).

The EM is then trained under the assumption that the channel switches between $K = 3$ states. It is noticeable from Table 4 that the 'blocking' state has the same statistical parameters in the model with $K = 3$ states as in the model with $K = 2$ states: the probability of loss in the blocking state is 0.9577; the channel blocks 1.9% of the time and the mean blocking period is about 1 second. It results from these remarks that the blocking state is preserved in the two models.

On the other hand, the second state of the model with $K = 2$ states is split into two states in the model of order $K = 3$: a first state with a probability of loss of 0.1028 and another state with a probability of loss of 0.2037. Note that the second state is very transitory; indeed the mean sojourn time in the intermediate state is 118 msec. In practice, this makes it impossible to base any decision to adapt the

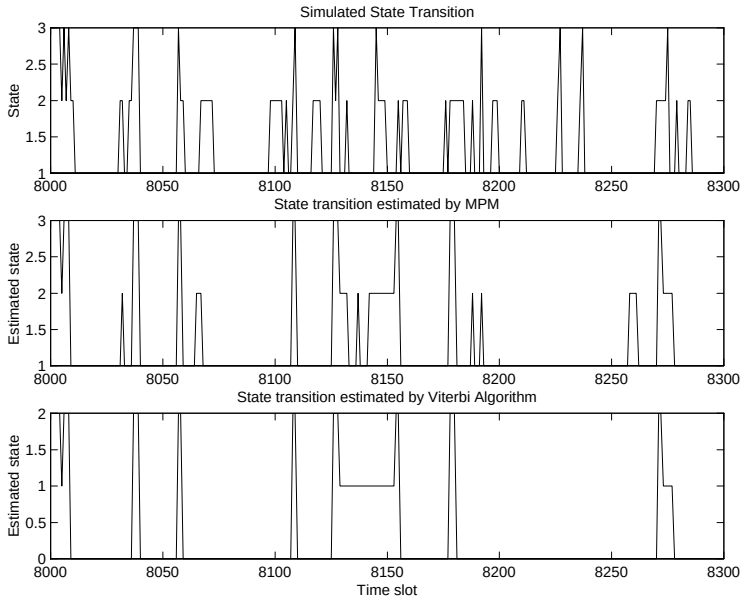


Figure 4: State transition estimation over the simulated packet loss trace.

Number of States	Mean Log-Likelihood	Vector $\hat{\mathbf{P}}$ State distribution $\hat{\pi}$	Estimated Transition Matrix Γ
1	0.5842	(0.14) (1)	
2	0.5106	(0.1109, 0.9466) (0.981, 0.019)	$\begin{bmatrix} 0.9991 & 0.0009 \\ 0.0468 & 0.9532 \end{bmatrix}$
3	0.5105	(0.1028, 0.2037, 0.9577) (0.90,0.081,0.019)	$\begin{bmatrix} 0.9626 & 0.0372 & 0.0002 \\ 0.4131 & 0.5776 & 0.0093 \\ 0.0048 & 0.0468 & 0.9484 \end{bmatrix}$
4	0.5105	(0.074, 0.124, 0.50, 0.948) (0.7122,0.22,0.06, 0.018)	$\begin{bmatrix} 0.8839 & 0.0705 & 0.0451 & 0.0006 \\ 0.1957 & 0.7051 & 0.0976 & 0.0016 \\ 0.6974 & 0.1880 & 0.1137 & 0.0009 \\ 0.0051 & 0.0344 & 0.0058 & 0.9547 \end{bmatrix}$

Table 4: Results of the EM algorithm applied to trace 1 for up to 4 states.

application on the detection of the channel being in this state.

When the EM is trained under the assumption that the order of the model is $K = 4$, the 'blocking' state is unchanged; the second state in the model of order $K = 2$ is split into 3 states with probabilities of loss 0.124, 0.50 and 0.948. But again the sojourn times in the intermediate states are too short to permit a decision that would be based on the detection of the channel being in one of the intermediate states. For that reason we think that a model with two states (blocking and passing) is sufficient in practice. Furthermore, the log-likelihood is no higher than the estimated entropy by more than 0.0001.

To conclude, a 2 state HMM is suitable for this trace. It is noticeable that following a procedure similar to that used in [16] would result in this case in a 10 state model. The estimated HMM model leads to an interpretation of the network behavior as the result of two different processes : one that generates a uniformly distributed 10% loss rate and another that results in link breakdown with mean duration of 1 sec, that occurs on average every 55 sec. The probability of being in a blocking state can be used here as a robust congestion indicator.

Results obtained from application of the estimation procedure to trace 2 are displayed in table 5. As transition matrices do not provide any intuition we have shown state distribution vectors which are directly derived from transition matrices.

Table 5 shows that the log-likelihood remains stable between a 3 and 4 state models. Moreover, the process will remain only 1% of time in the new added state in the 4 state model. For this trace a 3 state model is reasonable. The three states are here more difficult to interpret than for the first trace. The third state represents a loss burst of mean length around 10 (equivalent to 500 *msec*) occurring on average every each 25 sec. The two other states are representative of two network state (or regime). State 1 is a relatively quiet state with relatively low loss probability

State number	Mean Log-Likelihood	Vector $\hat{\mathbf{P}}$ / State distribution $\hat{\pi}$
1	0.6979	$\frac{(0.1883)}{(1)}$
2	0.64304	$\frac{(0.1613,\ 0.9928)}{(0.9720,\ 0.028)}$
3	0.6205	$\frac{(0.07,\ 0.20,\ 0.9968)}{(0.3148,\ 0.6582,\ 0.0270)}$
4	0.6205	$\frac{(0.07,\ 0.199,\ 0.55,\ 0.996)}{(0.3151,\ 0.6463,\ 0.011,\ 0.0270)}$

Table 5: Results of the EM algorithm applied to trace 2 for up to 4 states

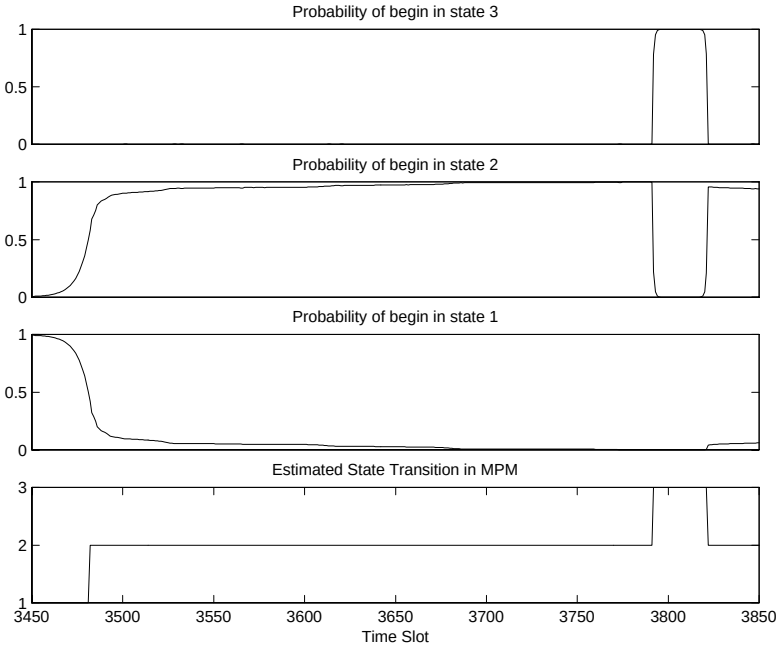


Figure 5: State probabilities for trace 2.

(0.07) and mean sojourn time around 10.5 sec (210 time slots). The second state is more perturbed with higher loss probability (0.20) and mean sojourn time around 2.6 sec (52 time slots). The sojourn time in this case, make adaptive decision possible as it provides sufficient time for the application to adapt to the state change. However, this is not always the case and some traces exhibit fast dynamics, with a sojourn time around 2 or 3 slots.

The state probabilities were also estimated for trace 2. These probabilities are shown in Fig. 5 for a 20 sec period (equivalent to 400 time slots). State changes are sharp and well separated. The sojourn time in each state is also long enough to make adaptive decision. This validates the utility of the state estimation procedure for adaptation purposes.

As shown in the above examples, the Hidden Markov Model (unlike to previous markovian model as proposed in [16]) provides some valuable information about the dynamics of the background network open/close process through the modelling of its sampled version. However, we have not investigated in this paper necessary conditions for characterizing completely the open/close process using the sampled version. This would be the subject of forthcoming papers.

We applied the above EM based estimation procedure to 36 sequences of 10 000 packets extracted from traces gathered over the Internet, including traces provided by the authors of [16] on their web site. 12 out of the 36 sequences show good adequacy with a single state model. These sequences were all gathered with a high interpacket delay ($\Delta \geq 100msec$) on links with a high level of multiplexing (such as the transatlantic link between France and the USA). 16 sequences show 2-state behavior, most of them having an interpretation similar to that provided for trace 1: one state with a uniform loss distribution and a second state representative of long loss burst. 6 sequences shows 3-state behavior, similar to trace 2. Only one trace exhibits a 4-state behaviour. 6 out of the 7 traces (including the 4 state trace) were gathered from transmissions between sites in France and sites in Germany. Our interpretation is that these traces do not have such a high level of statistical multiplexing as 1 state traces. This intuition is based on the

fact than there should not be as many competing flows between France and Germany compared to the transatlantic link. One of the 3 state traces was found on the web site of the authors of [16].

7. CONCLUSION AND PERSPECTIVES

In this paper we have applied the Expectation-Maximization methodology to the Maximum Likelihood inference of states of a network communication channel. We have also developed a procedure for choosing the correct number of states to be used to obtain a model as faithful and as simple as possible. This study shows that the HMM approach can be valuable in the context of IP performance evaluation.

The estimation procedure have two steps: the HMM calibration step that need a notable amount of data and should therefore be done offline, and the state estimation that tracks the state of the calibrated HMM using the observation. The first step needs around 10000 samples to give a stable entropy estimate as well as a HMM calibration. Actual implementation in Matlab environnement need less than one minute to calibrate the HMM using the EM algorithm. However an implementation in C++ will obviously achieve smaller estimation delay. The state tracking step can be done online in realtime with Marginal Posterion Mode. It can also be done with a constant delay by using the viterbi algorithm and a fixed look ahead window.

Another interesting problem that has been omitted from this paper concerns the tracking of non-stationarity and variation in the subjacent network channel by the HMM. Comparison of calibrated HMMs at different periods of time can be used to detect changes in the statistical behavior of an Internet channel. These two topics will be the subject of a forthcoming paper.

The HMM obtained by the procedure developed in this paper is also valid in adaptive applications. The state estimation algorithm clearly simplifies adaptation, however it induces an estimation delay that can drive the adaptation loop to an unstable behavior. The Evaluation of adaptive networking applications using the state estimation mechanism is another new perspective offered by this work. The integration of such an estimation mechanism in a video diffusion over the Internet is currently being studied.

8. ACKNOWLEDGMENTS

We would like to thank Don Towsley (Univ. of Massachuset) for his kind guidance and fruitful discussions. We are also grateful to anonymous referees for their helpful comments.

9. REFERENCES

[1] J. Andren, M. Hilding, and D. Veitch. Understanding end-to-end internet traffic dynamics. In *Proceedings of SIGCOMM' 98*, 1998.

[2] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics.* McGraw-Hill Publishing Company, May 1998.

[3] J. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive fec-based error control for internet telephony. In *Proceedings of IEEE INFOCOM*, pages 1453–1460, NY, March 1999.

[4] T. Cover and J. Thomas. *Elements of Information theory.* John Wiley and Sons, 1991.

[5] A. Feldman and W. Whitt. Approximating processes with superpositions of exponential processes. In *Proc. IEEE Infocom'97*, March 1997. Kobe, Japan.

[6] C.-C. Liu and P. Narayan. Order estimation and sequential universal data compression of a hidden markov source by the method of mixtures. *IEEE Trans. Info. Theory*, vol. IT-40:pp. 1167 – 1180, June 1994.

[7] Mathworks. Matlab.

[8] A. Moffat. Linear time adaptive arithmetic coding. *IEEE Trans. Info. Theory*, vol. IT-36:pp. 401 – 406, March 1990.

[9] V. Paxson. *Measurements and Analysis of End-to-End Internet Traffic.* PhD thesis, UC Berkeley, February 1997.

[10] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP performance metrics. *RFC-2330*, May 1998.

[11] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[12] S. M. Ross. *Introduction to Stochastic Dynamic Programming.* Academic Press, 1983.

[13] H. Schulzrinne, J. Kurose, and D. Towsley. Loss correlation for queues with bursty input streams. In *Proc. IEEE ICC '92*, pages 219–224, 1992.

[14] S. Vaton. *Modélisation statistique de trafic sur réseau local: application au contrôle dynamique de bande passante.* PhD thesis, ENST, Paris, 1998.

[15] M. Yajnik, J. Kurose, and D. Towsley. Packet loss correlation in the mbone multicast network. In *IEEE Global Internet Conf.,London,UK*, 1996.

[16] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. In *infocom*, New York, Mar. 1999.

[17] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Info. Theory*, vol. IT-23:pp. 337 – 343, May 1977.

[18] J. Ziv and N. Merhav. Estimating the number of states of a finite-state source,. *IEEE Trans. Info. Theory*, vol. IT-38:pp. 61 – 65, January 1992.

Forecasting Seasonal Traffic Flows

Lionel Fillatre Dmitry Marakov Sandrine Vaton
Computer Science Department
ENST Bretagne, Brest, France
Email: firstname.name@enst-bretagne.fr

Abstract

The problem of seasonal traffic flow forecasting is addressed in this paper. It is shown that SARIMA time series models are particularly relevant to model a seasonal traffic flow. The SARIMA process is represented in linear state-space form and classical Kalman recursions provide on-line forecasting values. Experiments on a real traffic flow validate the method by supplying accurate forecasts.

I. Introduction

Network monitoring and diagnosis are key elements to improving network performance. In particular, forecasting network traffic is essential for network dimensioning, load balancing and traffic engineering tasks. Given a traffic flow, it is desirable to propose an on-line algorithm which provides accurate forecasting values. A natural framework for this problem involves state-space models. This promising approach requires modeling traffic flow evolution and, surprisingly, classical approaches using times series theory have been studied very little. This work proposes to investigate the relevance of seasonal time series to model the traffic flow by assuming it is composed of a trend and a seasonal pattern with short time correlations.

II. Problem statement

Seasonal AutoRegressive Integrated Moving Average (SARIMA) processes have been introduced in the literature to model time series with trends, seasonal pattern and short time correlations. Let us denote y_t the number of bytes passing through the observed link during the time interval $[(t-1)\Delta\,;\,t\Delta]$ of duration $\Delta > 0$ for $t = 0, \pm 1, \ldots$ Let B be the backshift operator, whose effect on a time series y_t can be summarized as $(B^d y)_t = y_{t-d}$ for all integers d. As defined in [1], [2], a SARIMA $(p,d,q) \times (P,D,Q)_s$ process y_t verifies the equation:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D y_t = \theta(B)\Theta(B^s)e_t$$

where e_t is a white noise sequence. Here, ϕ, Φ, θ and Θ are polynomial functions of degrees p, P, q and Q respectively. The term $(1-B)^d$ is used to eliminate polynomial trends and $(1-B^s)^D$ is used to eliminate seasonal patterns with the period s. The multiplicative polynomial term $\phi(B)\Phi(B^s)$ models the autoregressive part of the time series and $\theta(B)\Theta(B^s)$ stands for the moving average part. The presence of polynomial terms in B^s enables the seasonal dependence in the traffic flow to be modeled. Box and Jenkins' methodology [2] is used to estimate all the parameters characterizing a given time series. Model orders are fixed by analyzing the autocorrelation and partial autocorrelation functions of time series: the results obtained are summarized in section III.

From a practical point of view, the relevance of the SARIMA process to model traffic flows depends on the ability of the model to forecast values of the traffic flow. SARIMA time series can be represented in several forms: here, only the linear state-space form is retained. Indeed, as discussed in [1], this model has many virtues and especially the availability of Kalman recursions. Let x_t be the stationary AutoRegressive and Moving Average (ARMA) process verifying:

$$x_t = (1-B)^d(1-B^s)^D y_t \quad \text{and} \quad \phi(B)\Phi(B^s)x_t = \theta(B)\Theta(B^s)e_t.$$

Then, the derivation of the state-space model from the SARIMA process leads to (see details in [1, p. 471]):

$$\left\{ \begin{array}{rcl} y_t & = & F\mathbf{x}_t + e_t \\ \mathbf{x}_{t+1} & = & G\mathbf{x}_t + He_t \end{array} \right. ,$$

where F, G and H are matrices with appropriate dimensions depending on the parameters of the SARIMA model. The state vector $\mathbf{x}_t$, defined as $\mathbf{x}_t = (x_{t-m+1}, \ldots, x_t, y_{t-d-sD+1}, \ldots, y_t)^T$ with $m = \max\{p+sP, q+sQ\}$, is

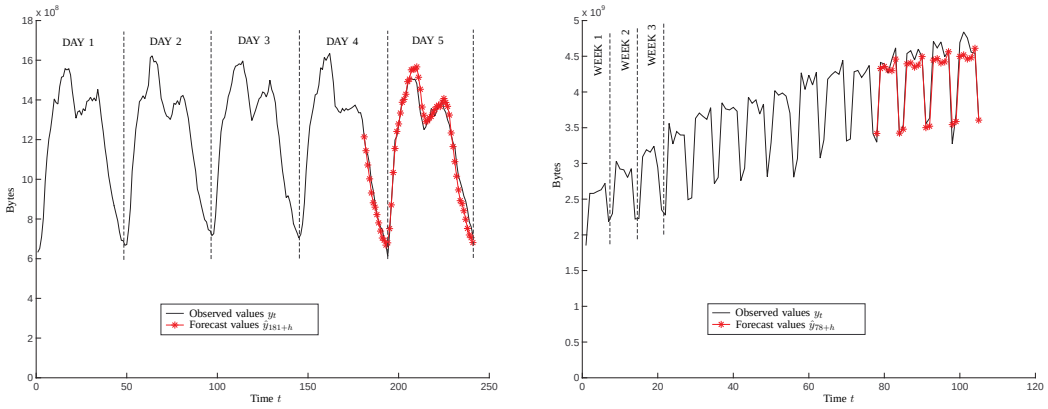
composed of 1) the past ARMA values $x_{t-m+1}, \ldots, x_t$ which stand for the stationary autoregressive $\phi(B)\Phi(B^s)$ and moving average $\theta(B)\Theta(B^s)$ part of the model and 2) past observations $y_{t-d-sD+1}, \ldots, y_t$ which stand for the non-stationary term $(1-B)^d(1-B^s)^D$.

Classical Kalman recursions are then used to forecast the traffic flow. In particular, at instant t, given the current and past observations $y_t, y_{t-1}, \ldots$, the Kalman h-step predictor gives the best linear prediction $\hat{y}_{t+h}$ of the values y_{t+h}. Kalman recursions permits an on-line calculation of $\hat{y}_{t+h}$.

III. Experimental results

To validate our model, a one-year traffic trace with a 5 minute observation interval is used. The trace was obtained from SNMP (Simple Network Management Protocol) reports of the routers freely accessible on the net and one of the most loaded routers representing the higher activity was selected. It can be seen from Figure 1.(a,b) that the observed traffic has very noticeable daily and weekly periodicities and a tendency of traffic growth over the time. These periodicities correspond to human activity, and the observed trend is the proof that the traffic demand is increasing. The higher the bandwidth of the link, the higher the human activity and the more and more noticeable the periodicities become.

To illustrate the theoretical developments, we consider two levels of data aggregation. The time origin is assumed to be 0 for each case. On the one hand, SNMP measurements are aggregated over half an hour and only five days are analyzed (see Figure 1.(a)). It can be shown that this traffic flow can be well approximated by a SARIMA $(1,1,1)\times(1,1,1)_{48}$ model and the noise e_t is approximately distributed as a zero mean white Gaussian noise with a well-estimated variance. Since the fourth day in the morning, the Kalman predictor algorithm is used to forecast the traffic for a long lead time corresponding to the afternoon of the fourth day and the whole of the fifth day (dotted line with stars in Figure 1.(a)). This simple model faithfully reproduces the seasonal pattern and supplies good forecasts. On the other hand, SNMP measurements are aggregated day by day for 15 weeks. Box and Jenkins' methodology [2] permits a SARIMA $(2,1,0)\times(1,1,0)_7$ model to be validated. The ability of this model to forecast values of the traffic flow for the last four weeks is shown on Figure 1.(b).



(a) (b)

Fig. 1. Traffic flow values measured every (a) $\Delta = 30$ minutes and (b) every day ($\Delta = 24$ hours).

IV. Conclusion

This paper shows that an Internet traffic flow can be well modeled by a SARIMA process by taking into account seasonal traffic patterns. In particular, Kalman recursions provides good forecasts. SARIMA models with multiple seasonal patterns will be studied in future work to consider traffic flows observed over a longer time period with a finer aggregation level.

References

[1] P. J. Brockwell and R. A. Davis, *Time series: theory and methods, second edition*. Springer, 1991.
[2] G. E. P. Box and G. M. Jenkins, *Time series analysis: forecasting and control, revised edition*. Holden-Day, 1976.

Annexe C : contributions en modélisation de trafic et évaluation de performances pour les réseaux EDGE/GPRS

1. DAHMOUNI Hamza, MORIN Bertrand, VATON Sandrine Performance Modelling of GSM/GPRS Cells with Different Radio Ressource Allocation Strategies. WCNC 2005 (IEEE Wireless Communications and Networking Conference), 13-17 March 2005, New Orleans, USA, 2005
2. DAHMOUNI Hamza, ROSSÉ David, MORIN Bertrand, VATON Sandrine Impact of Data Traffic Composition on GPRS Performances. 19th International Teletraffic Congress, august 29 - september 2 2005, Beijing, China

Performance Modelling of GSM/GPRS Cells with Different Radio Resource Allocation Strategies

Hamza DAHMOUNI and Bertrand MORIN
Methods and Tools for Traffic engineering
France Telecom R&D
Belfort, France
hamza.dahmouni@francetelecom.com

Sandrine VATON
Department of Computer Science
ENST Bretagne
Brest, France
sandrine.vaton@enst-bretagne.fr

***Abstract*—In this paper we present an analytical model for the performance evaluation of GSM/GPRS cells with different resource allocation schemes. The presented model is based on the Modified Engset Model with a finite number of users generating ON/OFF sessions in the cell. The closed formula following from the model is applicable for calculating different performance parameters like user throughput, blocking probability and radio resource utilization. Our study is focused on two main radio resource allocation strategies: Complete Partitioning and Partial Sharing.**

Keywords-GSM; GPRS; performance evaluation; Erlang-model; Engset-model.

I. INTRODUCTION

The main purpose of radio resource dimensioning is to define the necessary number of physical channels (number of TRXs) to guarantee an appropriate quality of service in the covered area. GSM operators have been dimensioning their networks for voice service in terms of offered voice traffic and blocking probability. The Erlang-B formula [19] developed in 1917 is the reference model for this system. This formula gives the proportion of calls that are blocked as a simple function of capacity and traffic intensity. It provides the relationship between offered traffic in Erlangs and the number of channels for a certain blocking probability (1% or 2% being the typical blocking values).

The GPRS network is designed for supporting several types of data traffic such as Wap, Web, E-Mail, etc... The main challenge in engineering of GSM/GPRS networks is the integration and support of a wide variety of applications such as voice and data services. These services require different qualities of service and the traffic corresponding to each data service is characterized by a typical ON/OFF process [5]. Unlike the GSM, the engineering of this network is more difficult because the traffic generated by data users is highly dependent on the application and is very bursty. Moreover, the possibility of sharing a single time-slot between several users complicates the analysis. Another major problem of GSM/GPRS operators is the choice of strategy to partition the available cell capacity between traditional GSM and new GPRS services.

Several research works have dealt with the performance evaluation of GPRS system. The major works in this field are based on simulations [17] [18]. Other works are based on analytical models using queuing theory and continuous-time Markov chains [10] [11] [12] [14] [15] [16]. All these models assume an infinite number of users in the cell, and thus model the arrival of traffic by a Poissonian distribution. Recently, in [1] [2] an analytical model based on discrete-time Markov chains has been proposed where a single type of traffic is considered and is assumed to be generated by a finite number of users. In this model, the allocation scheme adopted in the cell is Complete Partitioning and the data traffic is modeled by an ON/OFF process following a geometric distribution. In our study, we provide the same results using a *Modified Engset Model* where the traffic is modeled by ON/OFF process following a general distribution (Heavy tailed distributions). Furthermore, GSM/GPRS cells using Complete Partitioning and Partial Sharing with pre-emptive priority for the voice service are modeled, and important performance parameters like user throughput, blocking probability and radio resource utilization are deduced. For instance, extension of the model to take into account multiple traffic types has been obtained from our model.

The rest of the paper is organized as follows. Section II gives the main characteristics of GPRS systems that have an influence on the performance models. Section III describes the data traffic model. In Section IV, we present the mathematical description of the analytical models and the derivation of some interesting performance parameters. The last section draws a conclusion.

II. MAIN CHARACTERISTICS OF GPRS SYSTEMS

Unlike GSM that was designed for voice service and requires a circuit switching transmission mode, GPRS provides a packet switching transmission mode in GSM network. In this section we present the different radio access characteristics of this system that have an influence on the performance models:

- The basic transmission unit of a PDCH is called a radio block. The structure and the number of payload bits of a radio block depend on the coding schemes. The choice of the coding scheme mainly depends on the quality of the radio link, i.e. C/I. GPRS proposes four coding schemes (see TABLE I.):

TABLE I. TABLE GPRS CODING SCHEMES

GPRS Coding Schemes	CS-1	CS-2	CS-3	CS-4
RLC block radio (bytes)	23	33	39	53
Data rate: μ_{GPRS} (kbits/s)	9.05	13.4	15.6	21.4

- A GPRS user can transmit his data using simultaneously a number of PDCHs specified by its multi-slot class. Each PDCH can carry traffic for more than one connection simultaneously (a GPRS physical connection is called Temporary Block Flow (TBF)). Thus, the data transmission rate also depends on the number of GPRS users multiplexed on the same PDCH. The choice of the number of TBFs that a PDCH can have in downlink and uplink depends on the operator's choice. For example [10], in Ericsson technology up to 6 uplink and 8 downlink TBFs are allocated per PDCH. Siemens technology allocates up to 6 uplinks and 16 downlinks TBFs per PDCH. Every TBF is identified by a Temporary Flow Identity (TFI) with a 5 bits length, which limits the number of simultaneous GPRS connections to 32.
- Since data and voice users in the cell share the common transport media which has a limited capacity, the resources for GSM/GPRS traffic can be reserved statically or dynamically. However, a combination of both is also possible. Different cell capacity partitioning schemes for handling the transmission of GSM voice calls and GPRS data traffic can be defined. There are three main strategies to allocate such resources:
  - *Complete Partitioning (CP)*: the total cell capacity is divided into two fixed parts between voice and data traffic.
  - *Partial Sharing (PS)*: data users have their exclusive bandwidth but they can also use the available bandwidth of voice service with pre-emptive service priority for voice calls.
  - *Complete Sharing (CS)*: the total capacity is shared between the two kinds of users with pre-emptive priority for the voice service.

Among all these strategies, Complete Sharing was the first to be adopted for GPRS engineering (e.g., [12],[15]). This strategy is more conservative, since voice users are not affected by the introduction of a new data service. However, this choice obviously cannot prevent the transfer of any data packet being stopped when all the time-slots are occupied by the voice calls. Nowadays, the more popular cell capacity partitioning schemes are Complete Partitioning and Partial Sharing (e.g.,[1],[11],[14]. These two strategies guarantee an amount of bandwidth to be shared between all the active data users, since there is no technique that allows a QoS level for data traffic to be specified.

III. DATA TRAFFIC MODEL

Unlike GSM voice traffic, GPRS data traffic is elastic because the transmission rate is variable and it adjusts according to the available capacity in the network. The analysis of this traffic can be represented on three scales corresponding to three entities: the packets level where the elementary quantities are carried by the network, the flow level which is a concept closer to the application, a flow corresponding to a succession of packets, and the session level which is a succession of flows of the same application. The flows belonging to the same session are separated by think-times. The traffic generated by a session is usually characterized by an ON/OFF process where the ON-periods correspond to a flow or a succession of flows, e.g. send/reception of E-Mail messages or a Web pages load. The OFF-periods (think-times) correspond to the interval between the end of a ON-period and the beginning of the next one (see Figure 1.).

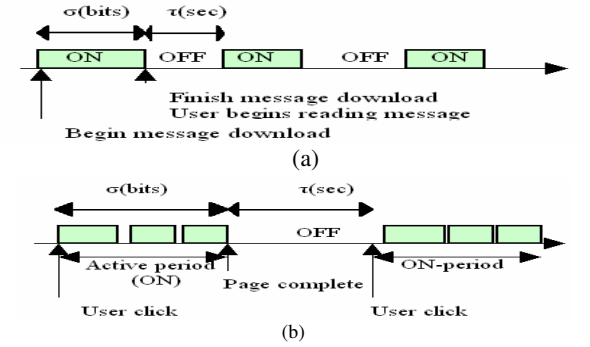


Figure 1. Example of (a) E-Mail exchange and (b) Web traffic load within a User session

Recently, analysis of live GPRS traffic measurements has revealed that packet-level traffic has two properties: Long-Range-Dependence (LRD) and self-similarity [9]. It proves also that it is very difficult to derive a packet-level traffic characterization, which is useful for performances modeling. Furthermore, the major performances modeling for data traffic invokes higher-level entities like the flow or the session.

The first flow-level model was given by L. Massoulie and J. Roberts [13], and is based on Processor Sharing scheduling. This model provides a simple and effective way of modeling the elastic properties of the data traffic and is described as follows: when there are $j \geq 1$ users present in the system, each user receives at a rate of $1/j$ times the service rate. An important property of this model is its *insensitivity*, i.e., the stationary distribution of the stochastic process describing the number of active users in progress does not depend on any traffic characteristics. It depends only on the mean flow size and the mean think-time duration. Therefore, the averages performances also depend only on these means.

On the other hand, measuring the size of documents such as Web pages, E-Mail messages, etc... shows that they present heavy-tail distributions, like Pareto, Weibull or Log-Normal (e.g., [8],[20]) with infinite variance. Due to the insensitivity property, these variances do not affect the stationary distribution, only the means of the file size and think-time duration are used in the formulas. As a consequence, we can present the Engset model [19] in terms of exponential ON/OFF distributions and apply the solution to any other distribution with the same mean.

IV. SYSTEM MODEL DESCRIPTION

The model that we develop is based on the Engset model that we adapt to take into account the variability of the transmission rate. Another adaptation of the model allows GPRS

users to have arbitrary ON/OFF distributions (not necessarily an exponential distribution). Similar Model adaptations have been used by [7] for modelling Web traffic over the Internet.

A. *Cells with Complete Partitioning Strategy*

1) Model assumptions: The allocation scheme adopted in the cell is a Complete Partitioning strategy. Among the T available time-slots, a number C_d is permanently reserved as PDCH channels for GPRS and $C_v = T - C_d$ time-slots are permanently reserved as TCH channels for GSM. Thus, the GSM and GPRS systems can be analysed separately.

Let N be the total number of GPRS users in the cell. We assume that all users are generating the same ON/OFF traffic (the same application: Wap, Web or E-Mail). All GPRS customers have the same multi-slot class "d+u" where d and u are the numbers of time-slots that can be simultaneously used for the downlink and uplink traffic (multi-slot capabilities). Most customers have $d = 4$ or 3 multi-slot capability in downlink and $u = 1$ or 2 in uplink. We focus only on resources employed for the downlink because the nature of data traffic is asymmetric and it is mostly downlink.

As mentioned in section II, the maximum number $n_{\max}$ of GPRS users in active transfer depends on the cell capacity, and it is limited by the maximum number m of users that can use a single time-slot and the number of TFIs per TDMA. Thus,

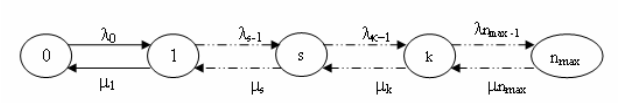
$$n_{\max}(C_d) = \min(N, 32, m\,C_d)\,. \qquad (1)$$

A fair sharing of bandwidth is used for all active users. So $s = \lfloor C_d / d \rfloor$ is the maximum number of active users such that the radio resource manager can allocate d time-slots for each one. If the number of users in active transfer is lower than s, each user has d time-slots, therefore each one receives a service with rate $(d\,\mu_{GPRS})$ where the data rate μ_{GPRS} depends on the coding scheme mentioned in section II. If not, each active user receives a service with the rate $(C_d\,\mu_{GPRS}/j)$ where j is the number of users in active transfer. Thus, this bandwidth sharing model can be recognised as a *Generalised Processor Sharing* (GPS) model considered by J.Cohen [3]. This model is similar to the Processor Sharing model except that the rate at which each user is served is a function of the number of users in the system, and also verifies insensitivity propriety.

2) Explicit adaptations of Engset Model: The basic adaptation of the Engset model is to include service rate reduction (GPS bandwidth sharing). There are N users in total, each user alternating between the ON and OFF periods. We assume that the ON-period size has a general distribution with mean $E[\sigma]$ and that the OFF-period duration follows a general distribution with mean $E[\tau]$.

The mean size of an ON-period is $E[\sigma]$, so when there is no radio resource congestion, the loading time is equal to $(E[\sigma]/d\,\mu_{GPRS})$. If there are more than s users in active transfer, all the C_d time-slots of the cell are used, then each user can be completed in an average time of $(E[\sigma]\,j/C_d\,\mu_{GPRS})$ where j is the number of users in active transfer at a given time. We assume that the bandwidth is shared equitably among active users.

Among the total users, a random number $J(t)$ are in the ON-state at time t, where $J(t) = 0,1,...,n_{\max}(C_d)$. This stochastic process describes the number of active users in progress and represents a finite state space. The insensitivity property allows us to consider an exponential distribution for an ON/OFF process. So we can think about $J(t)$ as a birth-and-death process with birth and death rates :



$$\lambda_j = (N - j)\frac{1}{E[\tau]}, \quad \text{for} \quad j = 0,1,...,n_{\max} - 1 \qquad (2)$$

$$\mu_j = \min(d, \frac{C_d}{j}) \cdot \frac{j \cdot \mu_{GPRS}}{E[\sigma]} \quad \text{for} \quad j = 1,...,n_{\max}. \qquad (3)$$

These adaptations determine our model. Similar Engset Model adaptations have been used by [7] for modelling Web traffic over the Internet, where the finite population in the model corresponds to a finite number of source alternates between "active" and "idle" periods.

Let $p(j)$ be the steady-state probability that j users are in active transfer. It is obtained by solving the steady-state equilibrium equations of the birth-and-death process, and it is described by the closed form here below:

$$p(j) = p(0)\frac{C_N^j}{\prod_{i=1}^{j} \min(d, C_d/i)}\rho^j \qquad (4)$$

where $\rho = (E[\sigma]/E[\tau]) \cdot \mu_{GPRS}^{-1}$ and $p(0)$ is obtained by the normalisation condition .

Notice that this steady-state distribution depends on the traffic parameters ($E[\sigma]$, $E[\tau]$) only through the ratio $A = E[\sigma]/E[\tau]$. Furthermore, it presents the same simplicity as an Erlang-B law. The use of this distribution allows us to determine analytically all the average performances of our system as a function of the load ρ (dimensionless parameter), the available cell capacity for data traffic C_d , the user capability d and the total number N of users.

3) Average performance evaluation: GPRS users perceive performance essentially through the average time necessary to transfer a document, so this time depends on the size of document and the average throughput obtained by the user. In the following, we evaluate performance in terms of throughput obtained by each user in active transfer (the size of an document devided on the transfer delay of that document) and blocking probability.

First, the average GPRS radio resource utilisation is obtained by [1]:

$$U_{C_d,N} = \sum_{j=1}^{n_{\max}} p(j) \frac{\min(jd, C_d)}{C_d}. \quad (5)$$

The main performance parameter of GPRS network engineering is the throughput obtained by each user. In our system, while the number of active users is *j*, each active user receives an effective bandwidth *r(j)*:

$$r(j) = \min(d, C_d / j)\ \mu_{GPRS} \quad \text{for} \quad j = 1, \ldots, n_{\max}. \quad (6)$$

The average overall throughput X_T of the resource is obtained as

$$X_T = \sum_{j=1}^{n_{\max}} p(j)\ j\ r(j). \quad (7)$$

This is the average rate at which the network delivers "bits" to all the active users, so these users receive a complete ON-period at an average aggregate throughput of $X_T / E[\sigma]$.

Let T_{on} be the average time taken by the network to complete the delivery of an ON-period. By Little's law, the average number $E[J]$ of active users equals the product of the average aggregate throughput with the average time T_{on}. Thus,

$$E[J] = (X_T / E[\sigma])\ T_{\text{on}}. \quad (8)$$

We finally calculate the average throughput $X_{C_d,N}$ obtained by each user in active transfer as

$$X_{C_d,N} = \frac{E[\sigma]}{T_{\text{on}}} = \frac{\sum_{j=1}^{n_{\max}} p(j)\ \min(jd, C_d)}{\sum_{j=1}^{n_{\max}} j\ p(j)}\ \mu_{GPRS}. \quad (9)$$

From this formula, we can derive the reduction factor (RF) of the throughput. So when there is no radio resource congestion, GPRS users get an average user throughput of $(d\ \mu_{GPRS})$. But, in GPRS a single time-slot can be shared by more than one user. The number of users sharing the same time-slot affects the average user throughput, therefore, the average throughput can be written as [15]:

$$X_{C_d,N} = d\ \mu_{GPRS}\ RF. \quad (10)$$

The value of *RF* varies between 0 and 1. It represents the cell load, so in a very saturated state it will be close to 0, while in low loaded cell it will be close to 1.

Another important performance parameter for GPRS dimensioning is the blocking probability. The arriving GPRS user is accepted to transmit/receive data if a sufficient number of free resources are available, i.e., the number of users in active transfer is lower than $n_{\max}$, otherwise it is blocked. Thus, the blocking probability is the probability that C_d time-slots are being used by $n_{\max}$ users among the other (*N-1*) users. It is given by the following expression:

$$B_{C_d,N} = 1 - \frac{1}{\rho} \frac{\sum_{j=1}^{n_{\max}} p(j) \min(jd, C_d)}{\sum_{j=0}^{n_{\max}} (N - j)\ p(j)}. \quad (11)$$

Formulas (4), (9) and (11) are also demonstrated by B. Baynat and P. Eisenmann [1] using discrete-time Markov chains while assuming that the ON-OFF periods follow a geometric distribution.

Finally, we give the relation that connects the blocking probability, the average throughput obtained by active users and the mean number of users in active transfer. Let Γ be the average number of new arrival users:

$$\Gamma = \sum_{j=0}^{n_{\max}} \lambda_j\ p(j) \quad (12)$$

The blocking probability can be expressed as follows:

$$B_{C_d,N} = \frac{\Gamma - X_T}{\Gamma} = 1 - \frac{X_T}{\Gamma} \quad (13)$$

Consequently, we obtain,

$$X_{C_d,N} = \frac{\Gamma\ (1 - B_{C_d,N})}{E[J]}. \quad (14)$$

This model has been implemented by a simple program written in C language. The input data is the number of fixed PDCHs (C_d), transfer rate for one PDCH (μ_{GPRS}), user capability (d), mean ON-period size ($E[\sigma]$) and mean think times ($E[\tau]$). The output data is the user throughput, blocking probability and the cell utilization.

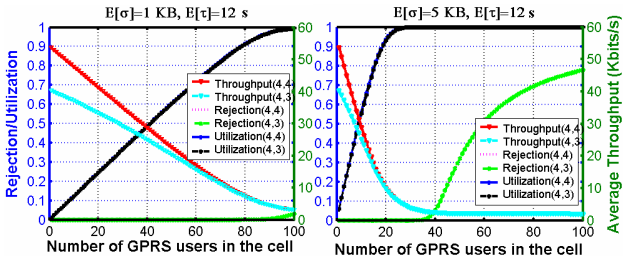


Figure 2. Average Throughput, Cell Utilisation and Blocking Probability versus the number of users in the cell for C_d=4 and user capability (d=4 or 3)

Figure 2 shows different curves for different performance parameters of GPRS traffic such as the average throughput, the blocking probability (rejection) and the average radio resource utilisation as a function of the number of GPRS users in the cell. These curves correspond to different values of ON-period sizes and OFF-period duration. We assume that up to 7 users can share a single time-slot ($m = 7$) and $\mu_{GPRS} = 13.4$kbits/s .

We can observe that as the number of GPRS users increases, the average throughput degrades rapidly (resp. slowly), blocking

probability and radio utilization increase rapidly (resp. slowly) for high (resp. low) load ρ, e.g. Web (resp. Wap). Furthermore, the multi-slot capability has a high influence on the throughput in low load cases. This is due to different levels of peak throughput when the number of users is small. On the other hand, for high cell loads ($U_{C_d,N} \sim 1$) the multi-slot capability impact is negligible.

Note that the two curves of blocking probability for ($C_d = 4, d = 4$ or 3) are equivalent. Thus is also true for the curves of average cell utilisation.

B. *Cells with Partial Sharing Strategy*

1) System model: Now, we consider that a cell adopts a Partial Sharing strategy. Among the T available time-slots, a number C_d is permanently reserved as PDCHs channels for GPRS (static PDCHs) and C_v time-slots are permanently reserved as TCHs channels for GSM, whereas the remaining $(T - C_d - C_v)$ time-slots are shared between the two services with pre-emption for the voice service (dynamic PDCH).

The choice of C_v and C_d influences the quality of service perceived by the users. The disadvantage of this strategy is obviously the increased number of blocked voice calls, due to the reduction in available time-slots. As a consequence, in order to guarantee a fixed voice call blocking probability, it is necessary to dimension the cell for reduced offered voice traffic and to guarantee a minimum QoS for data traffic. Therefore, a balance between the reduction of offered voice calls and the performance perceived by data users has to be defined by the operator.

We assume that the voice calls arrive as a Poisson process at rate λ_v and the call length is exponentially distributed with mean $1/\mu_v$. Then, the arrival-and-service behaviour for voice calls forms a $M/M/T-C_d/T-C_d$ queuing system.

2) Performance evaluation: The voice calls are independent of GPRS connections, but both types of traffic share the same air interface and use the same physical channels. On the other hand, the voice and data traffic evolve at different time scales, the service time of GPRS being rather shorter than that of voice calls. For example, as the time required to transfer a web page is about several seconds, it should be shorter than the mean call duration, which is about several minutes. As a consequence, the number of active GPRS users in progress evolves rapidly with respect to the process describing voice traffic. Therefore, we can suppose that between two variations of the number of voice calls, the number of data transfers reaches its stationary regime.

As an approximation, the decomposition technique [6] can be used to analyse the GPRS performance. The essential point for this technique is to use the voice call steady-probability to describe the interaction of voice calls with GPRS connections. Thus, the GPRS performances in a cell using Partial Sharing can be obtained by combining this distribution with the performances obtained in the previous subsection. This supposition is also used by [14] and [4] for modelling the integration of streaming and elastic traffic within an Internet network.

For voice calls, the steady-state probabilities are given by:

$$R(n) = \frac{\rho_v^n / n!}{\sum_{i=0}^{T-C_d} \rho_v^i / i!} \quad \text{for} \quad n = 0,1,...,T-C_d \qquad (15)$$

where $\rho_v = \lambda_v / \mu_v$.

Among $(T - C_d - C_v)$ time slots, those not used by the voice calls may be used for data traffic. Thus, the probability that ξ time-slots are available for the data transfer is equal to the probability that $(T - C_d - \xi)$ time-slots are used by GSM voice calls, i.e., $R(T - C_d - \xi)$.

A new arriving GPRS user is accepted only if the number of GPRS users in active transfer is below the maximum accepted number $n_{\max}$. Otherwise, the new arrival is blocked. Mentioning that $n_{\max}$ depends on the maximum number m of users that can use a single time-slot and the number of TFIs per TDMA, $n_{\max}(C) = \min(N, 32, m\,C)$ where C is the cell capacity available for GPRS traffic (see section II).

Combining (15) with (9) and (11), we can calculate the average throughput X_d obtained by each user in active transfer and the blocking probability B_d as follow:

$$X_d = \sum_{\kappa=0}^{T-C_d} R(\kappa)\, X(\min(T - C_v, T - \kappa)) \qquad (16)$$

$$B_d = \sum_{\kappa=0}^{T-C_d} R(\kappa)\, B(\min(T - C_v, T - \kappa)) \qquad (17)$$

where

$$X(C) = X_{C,N} \quad \text{and} \quad B(C) = B_{C,N}.$$

This model has been implemented by a program written in C language. The input data is the total capacity of the cell (T), the number of static PDCHs (C_d), the dynamic PDCHs number, the maximum offered voice traffic (ρ_v) that corresponds to 2 percent of blocking probability (2% GoS), transfer rate for one PDCH (μ_{GPRS}), user capability (d), average ON-period size ($E[\sigma]$) and average think times ($E[\tau]$). The output data is user throughput and the blocking probability.

Figure 3. shows the average throughput and the blocking probability as a function of the number of GPRS users in the cell. We consider a cell with 2 TRXs (total cell capacity is T=14 time-slots), one static PDCH ($C_d = 1$), three dynamic PDCHs (i.e. $C_v = 10$), up to 7 users GPRS can share a single time-slot ($m = 7$), and an offered voice traffic of 7.40 Erlangs, i.e., according to a blocking probability of 2%.

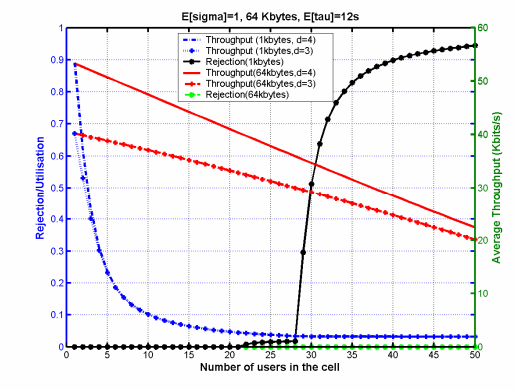


Figure 3. Average Throughput and Blocking Probability versus the number of users in the cell.

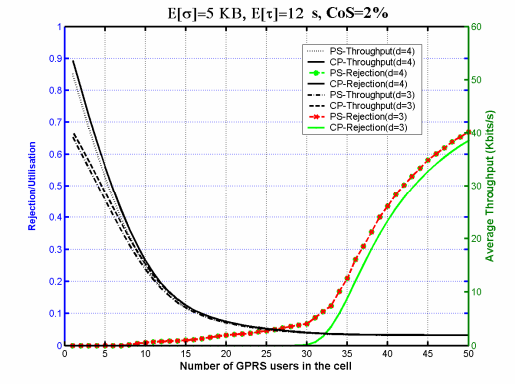


Figure 4. Comparing both PS and CP strategies.

Figure 4. depicts the comparison between the performance measures of two cells with different resource allocation strategies, and a maximum offered voice traffic corresponding to a 2% blocking probability ($\rho_v = 5.08$ Erlangs for a cell with CP and $\rho_v = 7.40$ Erlangs for a cell with PS strategy).

V. CONCLUSION

In this work, we have developed analytical models for the performance evaluation of GSM/GPRS cells with different resource allocation strategies. These models present a simplicity level comparable to the Erlang-B model. The proposed models allow us to better measure the performance of the system according to traffic load. The model has the advantage of considering the cell as a "closed" area with a finite number of users. That is in contradiction with the commonly existing models that have an "open" vision of the cell and assume that communication requirements are generated by an infinite number of users. And therefore assume that the arrival law of traffic follows a Poissonian distribution.

ACKNOWLEDGMENT

The authors would like to thank A. Proutière, T. Bonald, N. Benameur (France Telecom R&D/CORE) for the fruitful discussions. Further, we would like to acknowledge the support of A. Do, D. Rossé (France Telecom R&D/RESA). In particular, my grateful thanks to B. Baynat (LIP6), P. Eisenmann (Nortel) and to my PhD director Xavier Lagrange.

REFERENCES

[1] B. Baynat and P. Eisenmann, "Towards an Erlang-like Law for GPRS/EDGE Network Engineering". In proc. of IEEE ICC, Paris, France, June 2004.

[2] K. Boussetta, B. Baynat and P. Eisenmann, "Performance evaluation of GPRS/EDGE networks: A novel discrete-time Markov model", 5th World Wireless Congress (WWC 2004) San Francisco, USA - May, 2004.

[3] J.W. Cohen, "The Multiple Phase Service Network with Generalized Processor Sharing", Acta Informatica, volume-12, pp 245-284, 1979.

[4] F. Delcoigne, A. Proutière and G. Régnié," Modeling integration of streaming and data traffic", Performance Evaluation, 55 (3-4), pp 185-209, 2004.

[5] ETSI 3GPP, "Selection Procedure for the Choice of Radio Transmission Technologies of the Universal Mobile Telecommunications System UMTS", v3.1.0 edition, 3G TR 101-112), 11-1997.

[6] S. Ghani and M. Schwartz, "A Decomposition Approximation for the Analysis of Voice/Data Integration", IEEE Trans. Commun., vol.42, no.7, pp.2441-2452, 1994.

[7] D.P. Heyman, T.V. Lakshman and A.L. Neidhadt, " A New Method for Analysing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet", In proceedings of ACM SIGMETRICS, pp 24-38, 1997.

[8] J. Ho, Y. Zhu and S. Madhavapeddy, "Throughput and buffer Analysis for GSM General Packet Radio Service (GPRS)", In proc of IEEE WCNC, pages 1427-1431, New Orleans, USA, September 1999.

[9] R. Kalden and S. Ibrahim, "Searching for Self-Similarity in GPRS". The 5th annual Passive & Active Measurement Workshop, PAM 2004, France, April 2004.

[10] C. Lepschy, G. Minerva, D. Minervina and F. Pascali, "GSM-GPRS Radio Access Dimensioning", Vehicular Technology Conf., Vol. 2, pp 558 – 561, 2001.

[11] C. Lindemann and A. Thummler, "Performance Analysis of the General Packet Radio Service". Computer Network, 41/ 1-17, January 2003.

[12] M. Mahdavi, R. Edwards and P.A.Ivey," Performance Evoluation of Data Subsystem in GSM/GPRS Using Complete Sharing", Proc of London Communications Symposium, University Collage of London, 2001.

[13] L. Massoulié and J.W.Roberts, "Bandwidth sharing and admission control for elastic traffic", In Proc. ITC Specialist Seminar, Yokohama, 1998.

[14] S. Ni and S. Häggman, "GPRS Performance Estimation in GSM Circuit-Switched services and GPRS Shared Resource Systems", WCNC, Vol. 3, pp 1417-1421, 1999.

[15] S. Pedreza, J. Romero and Mûnoz, "(E)GPRS Hardware Dimensioning Rules with Minimum Quality Criteria", Vehicular Technology Conference, pp 391-395, Vol .1, 2002.

[16] T.S. Randhawa and R.H.S. Hardy, "Performance Analysis of Multi-Service Cellular Network with Mobile user", WCNC, Vol. 3, pp 1390 – 1397, 2000.

[17] P. Stuckman and O. Paul, "Dimensioning GSM/GPRS networks for circuit and packet switched services". Proceeding of the 10th Symposium on Wireless Personal Multimedia Communications, pages 597-602, Aalborg, Denmark, 2001.

[18] P. Stuckman, H. Fink and T. Bahls, "A Wap Traffic Model and its Appliance for the Performance Analysis of Wap over GPRS". In Proc. IEEE International Conference on third Generation Wireless and Beyond, Aalborg, Denmark, 2001.

[19] J. Virtamo "Queuing Theory", Complete lecture notes, Finland, 2002.

[20] W. Willinger, V. Paxson and M.S. Taqqu, "Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic". In A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions, Birkhauser Boston Inc., Cambridge, MA, 1998.

Impact of data traffic composition on GPRS performance

Hamza Dahmouni[1], David Rossé[1], Bertrand Morin[1] and Sandrine Vaton[2]

[1] France Telecom Research and Development, 6 avenue des usines - 90007 Belfort France
{hamza.dahmouni,david.rosse,bertrand.morin}@francetelecom.com,
[2] ENST-Bretagne, Dept. of Computer Science - CS 83818 - 29238 Brest Cedex 3, France
sandrine.vaton@enst-bretagne.fr

Abstract: The topic of this paper is the evaluation of the impact of data traffic composition on GPRS performance. A detailed statistical analysis of GPRS traffic per service at different levels has been carried on live measurements. A multi-service performance model is presented in order to derive simple and closed form formulas, associating capacity, traffic characteristics and performance. This performance model allows the analysis of the impact of a mix of the applications (WAP, Web, e-mail, etc…) on the quality of service QoS offered to GPRS users. It also gives the opportunity to evaluate the impact of the introduction of new services on the QoS offered to already existing services.

Keywords: traffic model, GPRS, live measurement, processor sharing, performance model

1. INTRODUCTION

The main challenge in the engineering of a multi-service mobile network is the integration and support of a wide variety of services such as WAP, MMS, Web, e-mail, etc. These data services require different levels of quality of service (QoS). Moreover the traffic corresponding to each service is highly bursty and can be characterized by a typical ON/OFF process [5]. The total traffic generated by data users is consequently highly dependent on the burstiness of each service. Optimal dimensioning of this multi-service network requires the knowledge of the traffic characteristics of each service.

The expected merge of the wireline and the wireless Internet implies that a dramatic change will occur in traffic composition within mobile networks. For example, Web traffic will probably overflow WAP traffic. Indeed, HTML compatible browsers will soon be widely available in mobile handsets. And the most popular wireless application WAP will emerge towards TCP, following WAP 2.0, and will show characteristics similar to the Web service.

In order to analyze how the content of the applications mix (e.g. WAP, Web) impacts the QoS of the GPRS networks, and also to evaluate the impact of the introduction of a new service on the performance of existing services, we have developed a multi-service performance model. This model relies on an ON/OFF model for each service. It requires a realistic characterization of the duration of the OFF-period and of the size of the ON-period.

That is why a traffic characterization has been carried out on live traces of the Orange GPRS network. The analysis of these live measurements was conducted at different levels (packet level and flow level) for various services supported by GPRS. Even though Internet traffic characterization has been a subject of interest for many years, very few results have been published to date on traffic characterization for wireless networks. Existing works

mainly focus on a packet level characterization of wireline Internet traffic, and especially on the well-known self-similar properties of the traffic on that level [11, 7]. Recently, much attention has been given to higher-level traffic characterization like flow [9] or session [8].

The goal of this paper is twofold. The first goal is to provide a realistic multi-service traffic model that is based on the statistical analysis of GPRS traffic traces measured on a live operational network. The second goal is to present an insensitive multi-service performance model for GPRS data traffic, that will be fed by the average multi-service traffic parameters (flow size and think-time averages) obtained by our statistical analysis. Moreover, a traffic characterization study was also the opportunity to analyze the impact of protocols (TCP, WTP, PPPoE, etc) on the characteristics of the traffic transmitted on the access network.

The rest of the paper is organized as follows. Section 2 deals with the composition of GPRS traffic in terms of transport protocols and in terms of service types. Traffic characterization at packet and flow level is developed in section 3: packet size distribution is studied for each service type, as well as flow size distribution (in bytes) and think-time distribution (in seconds) for each service type. In section 4 we present a multi-service performance model for GPRS and we give closed form formulas for the average performance for each service type. Section 5 is an evaluation of the QoS offered for each service type, under various traffic mix scenarios. Finally, section 6 gives a conclusion.

2. DATA TRAFFIC COMPOSITION IN GPRS NETWORKS

In this section, we present the different protocols employed by GPRS to ensure end-to-end applications. We also give traffic composition per TCP (or UDP) applications and IP protocol in terms of bytes, packets and flows in downlink and uplink traffic. The statistical results presented are realized on live GPRS measurements.

2.1. GPRS services and applications

GPRS can act as a packet wireless access to existing Internet services, notably using the WAP protocol, and offers new wireless services, such as MMS. Several protocols were defined to route data packets between the different network nodes according to the nature of their traffic. These protocols are WSP and WTP (over UDP) for WAP or MMS services; HTTP (over TCP) and DNS (over UDP) for Web navigation; POP3, SNMP (over TCP) for e-mail reception and SMTP for e-mail sending; FTP (over TCP) for file transfers. Unlike TCP, the UDP protocol ensures short transfer time but is less reliable. Several protocols were set up for GPRS systems to ensure the reliability of services using UDP (e.g. WSP, WTP,…)[10].

Among the new GPRS packet services, there is the packet switched (PS) WAP. This service has the same operation mode as the Web; it uses WML which is a markup language similar to HTML. Like the Web service, when a user wants to access a WAP site, he sends a WSP and WTP request which will be translated by a WAP gateway (located between the GGSN and the WAP/Web server) into a HTTP request. The server receives and interprets the request and then sends information directly either in WML to the gateway, or through an HTML filter which translates it into WML in order to be redirected to the user [10].

Unlike Web pages, WAP pages (called decks) are very small in size, about 1400 bytes (the WAP packet size is then usually limited to 1400 bytes). A deck comprises one or multiple objects (called cards), which have an average size of about 500 bytes.

2.2. Live traffic measurements

Few works have been published about multi-service wireless networks. Thus most existing performance models are based on wireline measurements. A characterization of

traffic patterns for each service is required in order to ensure an optimal dimensioning of GPRS networks. Understanding the nature of GPRS data traffic requires the knowledge of the protocols employed to support a given service.

That is why we have investigated GPRS data traces collected on the live Orange France GPRS network. The measurements were collected on two different G_b interfaces during a ten day period in July 2004 (about 5 hours per day, between 11 a.m and 4 p.m). The frames captured at the G_b interface contain the TCP or UDP/IP header information (packet size, port numbers, transfer direction, etc) and a timestamp which indicates the arrival time of packets with an accuracy of one millisecond.

This information allows the characterization of data traffic at different levels: packet, flow, ON-period and session. The main parameter of GPRS dimensioning is the Temporary Block Flow (TBF) throughput which is defined as the ratio between the data volume carried by a TBF and the time taken in transfer. In this study, we assume that the data transferred during an ON-period will be carried by a TBF, and the think-time is the interval time between two consecutive TBFs

A TBF corresponds to a group of packets such that the IP packet inter-arrival time is below the Keep Alive Timer (KAT). A more accurate distinction of the organization of the packets in a TBF can be achieved by separating the packets with the help of a 5-tuple {*IP protocol, IP source address, IP destination address, source port number, destination port number*}. Then a flow here will be a group of packets with an inter-arrival time below the KAT and with the same 5-tuple identifier.

WAP browsing and MMS services both use the WAP protocol. But as the recommended maximum size of a deck for the WAP is 1400 bytes and if we assume that a single deck is transferred per ON-period for WAP browsing, WAP browsing and MMS services can then be separated with the help of the ON period volume.

2.3. Protocol composition

Table 1 reveals the composition of the traffic per IP protocol in terms of bytes, flows and packets in uplink and downlink traffic. Unlike the wireline Internet where TCP is the most dominant traffic, Table 1 shows UDP's dominance over the mixed data traffic observed in GPRS. UDP averages at about 67% of the bytes, 65% of the packets, and 80% of the flows. TCP only represents about 30% of the bytes, 35 % of the packets, and 20% of the flows. The other IP protocols (IPv6, ICMP, etc), labeled "Other", represent a negligible percentage of the overall traffic (the port numbers only exist if TCP or UDP is used).

Table 1. Traffic composition by protocol

Protocol	bytes	byte DL	byte UL	flows	flow DL	flow UL	packets	packet DL	packet UL
TCP	31%	82%	18%	20%	49%	51%	35%	48%	52%
UDP	67%	79%	21%	80%	47%	53%	65%	39%	61%
Other	2%	65%	35%						

Like the wireline Internet, wireless traffic in terms of volume is asymmetric and is mainly downlink. The data loading direction is from the server towards the subscriber. About 80% of TCP/UDP traffic is in this direction. The uplink traffic is generally composed of requests, acknowledgements and control packets, and it represents about 20% of the volume.

TCP traffic is balanced on both links in terms of packets, but UDP traffic is predominant on the uplink. That is due to the use of higher-level protocols such as WSP and WTP on UDP. We also consider the DNS requests which are also transferred over UDP.

2.4. Application composition

As mentioned in the previous paragraph, UDP is dominant in GPRS data traffic. This is due to the high use of PS WAP and MMS services. Indeed, current mobile handsets mainly have a WAP browser and MMS capabilities. WAP browsers already existed in CSD and HSCSD modes, and MMS service usage is rapidly increasing. Popular services like logos and ring tones also use WAP. HTML and e-mail compatible mobile handsets as well as laptops equipped with a GPRS data card are not yet widely used; this may partly explain the relatively low usage of HTML and e-mail services in current GPRS networks. In the rest of the paper, we separate the WAP traffic between WAP browsing and all other services, such as MMS, logos and ring tones, carried by the WAP protocol (also using port 9201).

Table 2. Traffic asymetry by application

Application	byte DL	byte UL	flow DL	flow UL	packet DL	packet UL
WAP	72 %	28 %	50 %	50 %	38 %	62 %
Other WAP	86 %	14 %	50 %	50 %	40 %	60 %
HTTP	82 %	18 %	50 %	50 %	48 %	52 %
E-mail send	6 %	94 %	42 %	58 %	47 %	53 %
E-mail recv	91 %	9 %	48 %	52 %	50 %	50 %
Other	78 %	21 %	40 %	60 %	50 %	50 %

Table 2 shows the traffic composition per application in terms of volume and packets in both transfer directions. We present only the most usual services like WAP, Web and e-mails. The services named "Other" group all the other services. It is observed that WAP and Web services have a different asymmetry for the volume of traffic and the number of packets.

3. TRAFFIC CHARACTERIZATION

The analysis of data traffic can be achieved on three scales corresponding to three entities: packet-level which is the elementary quantity carried by the network, flow-level which is a sequence of packets associated with a single instance of an application, and session-level which is a succession of flows for a given application. Flows belonging to two consecutive ON-periods are separated by a think-time. Figure 1 shows the different time scales of our statistical analysis. The mini-OFF time presents the inter-flow time in the ON-period which is limited by the KAT that is used to artificially maintain a TBF allocated for successive data transmissions.

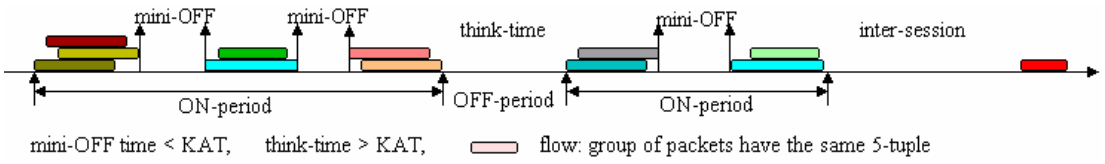


Figure 1. Different time scales of data traffic

3.1. Packet-Level Characterization

IP packet size in data traffic is very changeable. It is different for different transport protocols and according to the transfer direction. Figure 2 depicts the cumulative distribution function (CDF) of the packet size in the GPRS traffic studied. As can be seen, the TCP "downlink" curve has typical distinct steps at 40, 576 and 1500 bytes. It is similar to the curves obtained for wireline networks. However, the UDP "downlink" curve is very smooth

and has a typical step at 1420 bytes, which is the MTU size of WAP packets. To perform an analysis for each service, we present figure 3 giving the packet size CDF by application.

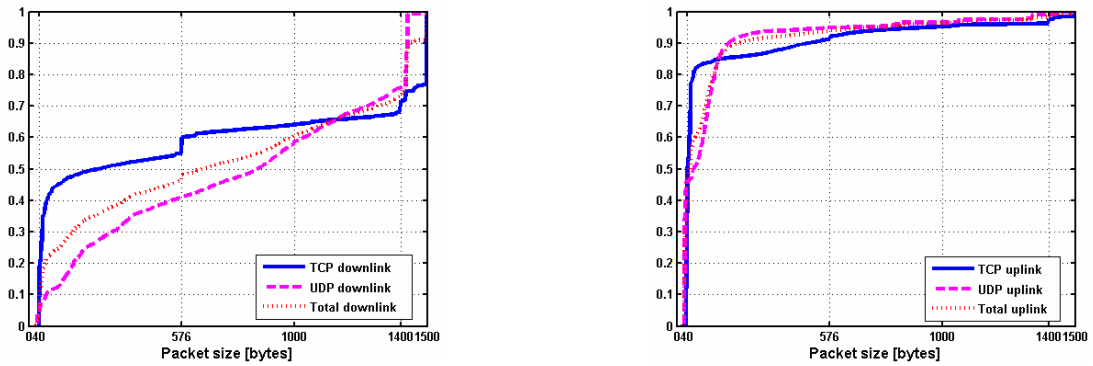


Figure 2. Packet size CDF per protocol in downlink (left) and uplink (right) traffic

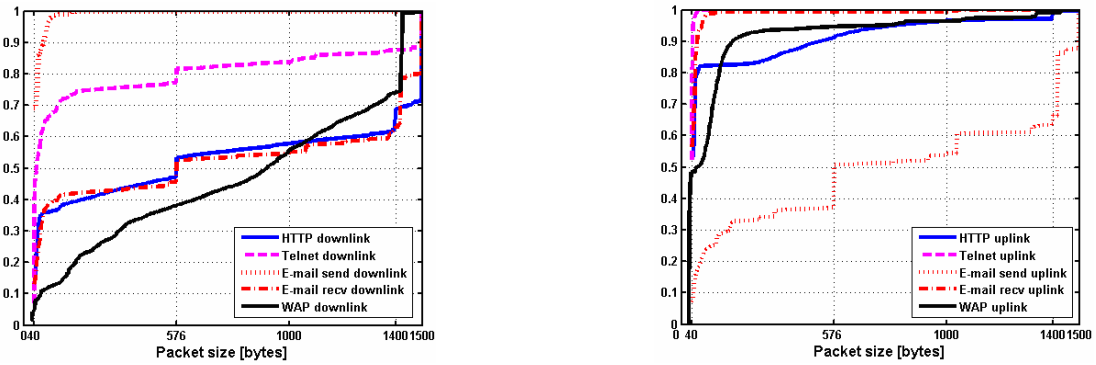


Figure 3. Packet size CDF per application in downlink (left) and uplink (right).

Downlink traffic: for TCP Applications, we observe a distinct step at around 40-44 bytes. These correspond to acknowledgement and control packets like SYN, FIN, RST. A second step at around 576 corresponds to the default TCP MTU size. Several packets have a size between 1420 and 1492 bytes. This last value corresponds to the PPPoE MTU size. The last step at 1500 bytes corresponds to the Ethernet MTU size. This size is adapted for Ethernet and xDSL networks. It is also adapted to the peak throughput offered by GPRS networks.

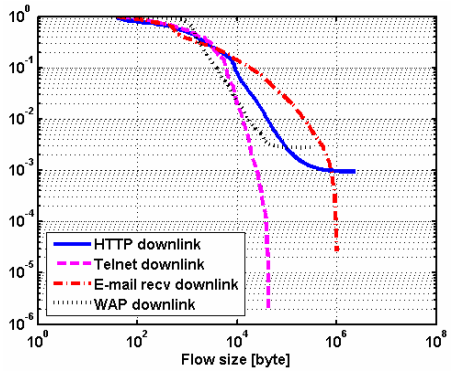
For UDP applications like WAP/MMS, packet sizes are smoothly distributed with a peak for 1400 bytes. This value corresponds to the default size of the WSP SDU. This size also corresponds to a commonly recommended maximum size for decks.

Uplink traffic: for all services, most IP packets are small sized. The packets essentially consist of acknowledgements, controls and requests. The remaining packets correspond to upload traffic for both UDP and TCP. For e-mail traffic we observe a peak at around 1006 bytes corresponding to the SLIP MTU size (Figure 3).

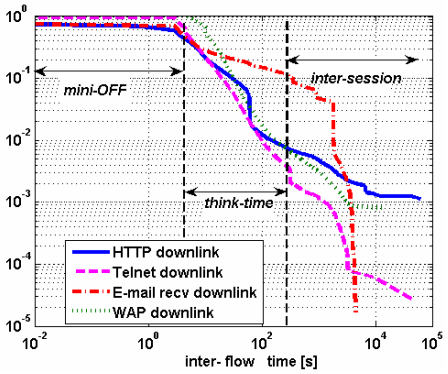
3.2. Flow-Level Characterization

Figures 4 (a) and (b) depict the complimentary CDF (CCDF) of flow size and the inter-flow duration per service respectively. As can be seen in figure 4(a) the distributions of flow size in bytes of HTTP and WAP applications (e-mail and Telnet applications respectively) have similar tail distribution. Several works have characterized these tails by heavy-tailed distributions like log-normal, Weibull and Pareto [9]. In figure 4(b), we can identify the mini-OFF, think-time and inter-session time scales. The inter-flow duration is defined as the

duration between two flows. It is called mini-OFF if the duration is lower than the KAT. If this duration is greater than the KAT then we call it think-time. Lastly, if it is superior to 180s we consider that it is an inter-session. As can be seen from the curves, the distributions of flow size and inter-flow time cannot be represented by a single law. A study in order to find a mixture of laws fitting the whole distribution is in progress.



(a)



(b)

Figure 4. Log-scale plots of (a) flow size CCDF and (b) inter- flow time CCDF.

Figure 5 shows the linear scale plot of WAP and HTTP flow size CDF. As mentioned in §2.4, we distinguish between WAP browsing and all other services carried by the WAP protocol such as MMS, logos, etc. A major part of the WAP flows are small sized, between 30bytes and 1kB which correspond to WAP browsing service. For HTTP and other WAP applications, flow sizes mostly range between 1kB and 10kB. Figure 6 shows the linear scale plot of inter-flow time CCDF for WAP and HTTP applications. As can be seen a large range of times are small, in particular for the Web where the flows of a Web page arrive in parallel.

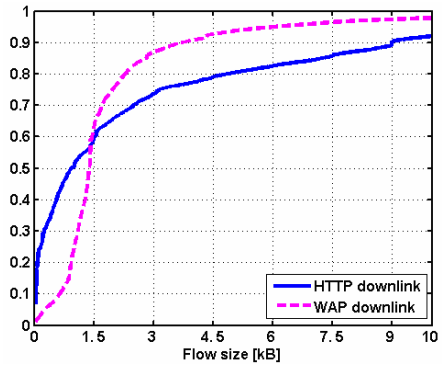


Figure 5. WAP and HTTP flow size CDF

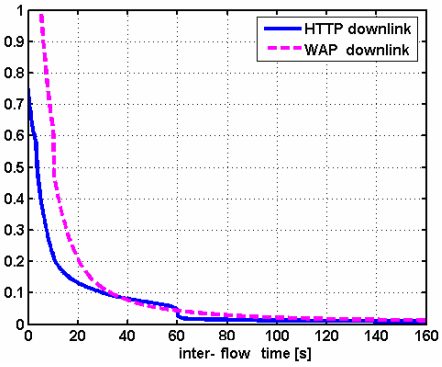


Figure 6. Inter-flow time CCDF

The model developed in [3][4] is insensitive, hence the average performance does not depend on detailed traffic characteristics but only on the average traffic parameters. Thus, the insensitivity property allows us to deal only with the characterization of the average file size and the think time duration. The distribution does not have any impact on the average performance perceived by the end user (see section 4). In table 3, we give details of average traffic parameters for the most commonly-used services, WAP browsing and the Web. Thus, the ON-period for the Web (resp. WAP) has an average size around 17.5kB (resp. 1.1 kB). The think-time was set to 14s (resp. 16.5s). These results are close to results published in [6].

Table 3. Web and WAP browsing parameters

Web Parameters	Average
Flow size [byte]	4700
Flows per page	3.7
Pages per Session	6.5
Page inter-arrival time [s]	14

WAP Parameters	Average
Card size [bytes]	500
Decks per session	16.5
Deck size [bytes]	1100
Deck inter-arrival time [s]	16.2

4. PERFORMANCE MULTI-SERVICE MODEL FOR GPRS

A multi-service model has been proposed in [1] for modeling high speed IP access links. A similar approach is used here to obtain a closed form formula to analytically evaluate different performance indicators for GPRS networks.

4.1. Assumptions

In a GPRS system, the basic transmission unit of a PDCH is called a radio block. The structure and the payload size of a radio block depend on the coding scheme. Several PDCHs can be simultaneously used by a mobile to transfer data depending on the multi-slot class of the mobile handset. Nevertheless, each PDCH can carry traffic for more than one connection simultaneously (a GPRS physical connection is called a TBF). Thus, the data transmission rate also depends on the number of GPRS users multiplexed on the same PDCH.

The number of TBFs that a PDCH can have in downlink and uplink depends on the operator's choice. Each TBF is identified by a TFI of 5 bits length which limits the number of simultaneous GPRS connections to 32 per TDMA. Thus, the maximum number $n_{\max}$ of data users in active transfer (active TBF) depends on the number of channels (C_d) reserved as PDCH for GPRS, and is limited by the maximum number m of users that can use a single time-slot and the number of TFIs per TDMA (i.e., $n_{\max} = \min(32, m\, C_d)$).

We assume that a fixed number N of GPRS users in the cell generates ON/OFF traffic, and that all GPRS customers have the same multi-slot capabilities "d+u" where d and u are the numbers of time-slots that can be simultaneously used for the downlink and uplink traffic. Most customers have "d = 4 or 3" multi-slot capability in downlink and "u = 1 or 2" in uplink. We focus only on resources employed for the downlink because the nature of data traffic is asymmetric and it is mostly downlink (Table 1 § 3.3).

As the maximum number of time-slots per user is d, $s = \lfloor C_d / d \rfloor$ is the maximum number of active users if the radio resource manager allocates d time-slots for each one. If the number of users in active transfer is lower than s, each user has d time-slots. Therefore, each one receives a service with rate ($d\ \mu_{GPRS}$) where the data rate μ_{GPRS} per time-slot depends on the coding scheme used (μ_{GPRS} (kb/s) = 9.05 for CS1, 13.4 for CS 2, 15.6 for CS3 and 21.4 for CS4). A fair sharing of total bandwidth is applied for all active users in transfer when the maximum number of time-slots cannot be allocated to all active users. When the number of active users is greater than s, then each active user receives a service with the rate ($C_d\, \mu_{GPRS} / j$) where j is the number of users in active transfer.

This bandwidth sharing model can be recognized as the *generalized processor sharing* model considered by J.Cohen [2]. An interesting result of this model is that the stationary distribution of the stochastic process describing the number of active users in progress does not depend on any traffic characteristics except the averages; it depends only on the average of the ON-period and of the OFF-period. This is known as the *insensitivity property* which allows us to express the average performance only in terms of the average parameters.

4.2. Multi-service model

In [3] we developed a basic model for the performance evaluation of GSM/GPRS systems. In this first model all GPRS users use the same application. In this section, we extend our basic model in order to take into account different types of traffic. In principle, several types of traffic such as WAP, Web, MMS, e-mail, etc. with different ON/OFF distributions have to be supported by GPRS networks. In what follows, M will be the total number of different traffic types. Each GPRS traffic type has its own general distribution. $E[\sigma_k]$ will be the average value for the ON-period size and $E[\tau_k]$ will be the average value for the OFF-period duration for a traffic of type-k, $k = 1,\ldots,M$. Because of the insensitivity property, the results are the same even if each traffic type has exponential distributions with mean 1 for the ON-period size and with mean $A_k^{-1}=E[\tau_k]/E[\sigma_k]$ for the OFF-period duration.

We assume that there is a fixed number N_k of GPRS users generating traffic with type-k ($N=N_1+\ldots+N_M$). Under the assumptions of section 2.1, the system can be modeled as an M-dimensional birth-and-death Markov process, with vector $\vec{j}=(j_1, j_2, ..., j_M)$ representing the state of the system, i.e., the number of active users for all service types in the cell at a given time. The steady-state probability of this system is obtained by solving the steady-state equilibrium equations, and it is given by the following closed product-form [4]:

$$p(\vec{j}) = p(\vec{0})\prod_{k=1}^{M} C_{N_k}^{j_k}\,\rho_k^{\,j_k}\prod_{i=1}^{j_1+\ldots+j_M}\frac{i}{\min(id, C_d)}, \qquad (1)$$

for all "feasible" states $\vec{j}$ in $\Xi=\left\{\vec{j}=(j_1,\ldots,j_M)/\quad 0\le j_i\le N_i,\ i=1,\ldots,M \text{ and } 0\le j_1+\ldots+j_M\le n_{\max}\right\}$, $\rho_k = A_k\mu_{GPRS}^{-1}$ and $p(\vec{0})$ is obtained by the usual normalization condition. Note that this distribution depends on the traffic parameters ($E[\sigma_k],E[\tau_k]$) only through the ratio $E[\sigma_k]/E[\tau_k]$.

4.3. Average performance evaluation

The GPRS user perceives performance essentially through the average time necessary to transfer a document, this time depending on the size of the document and the average throughput obtained by the user. In this section, we evaluate performance in terms of throughput available for each user in active transfer and blocking probability.

Let $T_{ON,i}$ be the average time taken by the network to complete the delivery of an ON-period of type-i. According to Little's law, the average number of active users ($E[J_i]$) is equal to the product of the average aggregated throughput (X_{T_i}) with the average time $T_{ON,i}$. Thus,

$$T_{ON,i} = \frac{E[J_i]\,E[\sigma_i]}{X_{T_i}} = \frac{\sum_{\vec{j}\in\Xi} j_i\ p(\vec{j})}{\sum_{\vec{j}\in\Xi-\{\vec{0}\}} p(\vec{j})\ j_i\ \min(d, C_d/j_1+\ldots+j_M)\mu_{GPRS}}E[\sigma_i], \qquad (2)$$

and therefore, the average throughput X_{C_d,N_i} obtained by each type-i user in active transfer is:

$$X_{C_d,N_i} = \frac{E[\sigma_i]}{T_{ON,i}} = \frac{\sum_{\vec{j}\in\Xi-\{\vec{0}\}} p(\vec{j})\ j_i\ \min(d, C_d/j_1+\ldots+j_M)}{\sum_{\vec{j}\in\Xi} j_i\ p(\vec{j})}\mu_{GPRS}. \qquad (3)$$

Another important performance parameter for GPRS dimensioning is the blocking probability. The arriving GPRS user is accepted to transmit/receive data if a sufficient number of free resources are available, i.e., the number of users in active transfer is lower than n_{max} , otherwise it is blocked. Thus, the blocking probability is the probability that C_d time-slots are being used by n_{max} users among the other ($N-1$) users. It is given as follows:

$$B_{C_d,N_i} = \frac{(N-n_{\max})p(n_{\max})}{\sum_{\vec{j}\in\Xi}(N_i-j_i)\ p(\vec{j})} = 1-\frac{1}{\rho_i}\frac{\sum_{\vec{j}\in\Xi-\{0,0\}} p(\vec{j})\ j_i \min(d, C_d/j_1+...+j_M)}{\sum_{\vec{j}\in\Xi}(N_i-j_i)\ p(\vec{j})}. \quad (4)$$

Consequently, we obtain the simple and analytical formulas (3) and (4) that could be used for GPRS dimensioning. These formulas request only the characterization of the average ON-period size and of the think-time duration. The distribution does not have an impact on the average performances perceived by the end user.

5. GPRS PERFORMANCE EVOLUTION

Today, a major part of the traffic carried by GPRS is generated by WAP browsing and Web navigation (Table 1 § 2.3). This network can be dimensioned with the performance indicators (formulas (3) and (4)). In this section we evaluate QoS indicators for WAP and Web services in one radio cell (using a single Coding scheme CS-2, 4 fixed PDCHs and m =7) as a function of the total numbers of users in the cell. Figures 7 and 8 show respectively the QoS indicators for a simplified present mix scenario (WAP browsing, Web), 75% of WAP and 25% of Web, and in a possible dramatic change of this composition, e.g., traffic containing 34% of WAP and 66% of Web. Since WAP applications will emerge towards TCP-applications using WAP 2.0, they should show characteristics more similar to the Web.

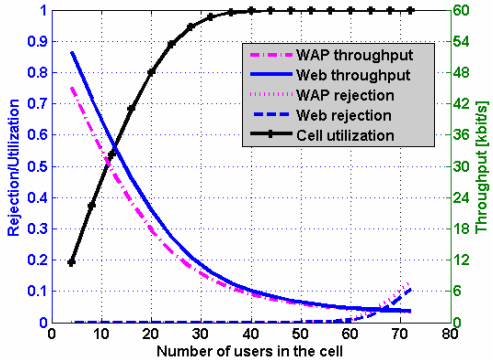


Figure 7. Mix of (WAP (75%), Web (25%))

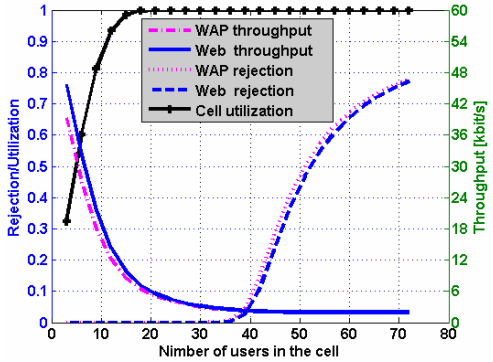


Figure 8. Mix of (WAP (34%), Web (66%)).

From the curve of the first mix scenario, we can observe that as the number of users increases, the average performance slowly degrades. In fact, the total traffic is mainly generated by the WAP users who have a low data volume (ON-period size is about 1kB) to transfer. Thus the TBF duration is short; hence the WAP user finishes his transfer rapidly and releases the radio resource (TBF) rapidly for another user. As a consequence, the blocking probability and radio resource utilization increases slowly when the number of users increases. The average throughput decreases but is not strongly impacted by WAP load (ρ) because the total traffic load is closer to the Web load.

In the second mix scenario, the total traffic is mainly generated by Web users who have a high data quantity to transmit (ON-period size is about 18kB). Thus, the TBF duration is much longer; the Web user occupies the radio resource for a long time. As a consequence, the average performance degrades rapidly when the number of users increases. In this case, the service load has negligible impact on the performance because the radio resource is saturated rapidly; hence all users have the same throughput. This is due to the *GPS* sharing bandwidth.

We can conclude that the system saturation level is strongly dependent on the presence of services with a high ON-period size (e.g. Web service, FTP or a new service like P2P). For several active sessions, the available PDCHs are fully loaded, thus the average throughput decreases rapidly as the number of users' increases. Furthermore, the presence of a service with a high load ρ has a great influence on the performance of other services available in the system, in particular, the services consuming less bandwidth like WAP services.

6. CONCLUSION

In this work we have presented the traffic composition of today's GPRS networks in terms of transport protocol and service types both for uplink and downlink. A realistic traffic characterization per service at different levels has been provided on live measurements of an operational network. We have also presented an insensitive multi-service performance model, which provides a simple relationship between offered data traffic, cell capacity and QoS parameters. Since wireless protocol and services continue to evolve, such a model is essential for an effective optimization of radio resource allocations. It makes it possible to evaluate the users' performances when a dramatic change occurs in the traffic composition, and to predict the impact of the introduction of a new service on the QoS offered to the existing services.

ACKNOWLEDGEMENT

The authors would like to thank P. Olivier and T. Bonald (France Telecom research and development) for many fruitful discussions as well as the persons in Orange France who provided the necessary measurements from the network.

REFERENCES

1. T. Bonald, P. Olivier and J. Roberts, 'Dimensioning IP Access Links Carrying Data Traffic', Annals of telecommunications, vol. 59, n°11-12, 2004.
2. J.W. Cohen, 'The Multiple Phase Service Network with Generalized Processor Sharing', Acta Informatica, volume-12, pp 245-284, 1979.
3. H. Dahmouni, B. Morin and S. Vaton and, 'Performance Modelling of GSM/GPRS Cells with Different Radio Resource Allocation Strategies', IEEE WCNC, New Orleans, 2005.
4. H. Dahmouni, D. Rossé, B. Morin and S. Vaton, 'Analytical Model for Performance Evaluation of GPRS/EDGE Multi-service Networks', 7th IFIP International Conference on Mobile and Wireless Communications Networks, Marrakech, Morocco, 2005.
5. ETSI 3GPP, 'Selection Procedure for the Choice of Radio Transmission Technologies of the Universal Mobile Telecommunications System UMTS', 3G TR 101-112), 1997.
6. C. Hoymann and P. Stuckmann, 'Traffic Engineering for the Evolution of GPRS/EDGE Networks', IEEE Vehicular Technology Conference, Vancouver, Canada, 2002.
7. R. Kalden, S. Ibrahim: 'Searching for Self-Similarity in GPRS', PAM 2004: 83-92.
8. R. Kalden, T. Varga, B Wouters and B. Sanders, 'Wireless Service Usage and traffic Characteristics in GPRS networks', proc ITC' 18, Berlin, 2003.
9. P. Olivier and N. Benameur, 'Flow Level IP Traffic Characterization', proc ITC'17, 2001.
10. T. Varga, B. Haverkamp and B. Sanders, 'Analysis and Modelling of WAP Traffic in GPRS Networks', ITC'16 Specialist seminar, Antwerp, 2004.
11. W Willinger, V. Paxson and M.S. Taqqu, 'Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic', A Practical Guide to Heavy Tails, Cambridge, 1998.

Annexe D : modèles économiques pour les services avec QoS garantie dans un contexte inter-domaines

1. AMIGO Isabel, BELZARENA Pablo, LARROCA Federico, VATON Sandrine, Network bandwith allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations, 7th International ICQT Workshop on Advanced Internet Charging and QoS Technology, Paris, octobre 2011
2. AMIGO Isabel, BELZARENA Pablo, VATON Sandrine, On the problem of revenue sharing in multi-domain federations, IFIP Networking 2012 Conference, Prague, mai 2012

Network Bandwidth allocation with end-to-end QoS constraints and Revenue Sharing in Multidomain Federations

Isabel Amigo[1],[2], Pablo Belzarena[1], Federico Larroca[1], and Sandrine Vaton[2]

[1] Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.
[2] Telecom Bretagne, Brest, France.

Abstract. Internet is evolving, traffic continues to grow, new revenue sources are sought by Network and Service Providers. Value added services with real time characteristics are likely to be common currency in the near future. Quality of Service (QoS) could allow Application/Service Providers (APs) to offer better services to the end users. At the same time, all actors claim for a fair distribution of revenues. Inspired by this scenario, we propose a complete framework for selling interdomain quality assured services, and subsequently distributing revenues, in an Autonomous System (AS) association context. We state the problem as a network utility maximization problem with QoS constraints and show that a distributed solution can be carried out. In order to fairly share the resulting revenue we study concepts from coalitional game theory and propose a solution based on the Shapley value and statistics on the revenues. Simulations of the whole proposal are shown.

Keywords: Auctions, QoS, Shapley Value.

1 Introduction

Internet traffic is likely to continue increasing in a non-stop fashion. Recent studies [2] show that not only the tendency is to increase in amount but in quality requirements as well, since the applications which are envisioned to have the greatest increase are those with real time characteristics.

Nowadays, the focus of telecommunication market is on best effort content and in order to meet customer expectations telecommunications companies are forced to invest in capacity, without getting sufficient return on these investments to have sustainable businesses. The ever evolving features provided by the handset terminals, and the growing number of connection capable equipments, constitute more evidence in favor of the forecast of Internet traffic increase.

Moreover, emerging technologies such as telepresence or cloud computing not only generate large volumes of traffic with real time requirements, but are also used to interconnect sites around the globe. As a consequence, in addition to a QoS capable network, this kind of services require an end-to-end (end-to-end) QoS enabled chain crossing heterogeneous carrier networks [15].

In this scenario, nowadays Internet business rules for domain interconnection may not be able to provide a sustainable economy for all actors in the value chain (Application Providers, Network Service Providers, etc.). Indeed, these rules (peering agreements) are not aware of the QoS capabilities of the domains and most of them are based on a traffic-symmetry premise that may no longer be valid in evolving services (for instance HD video on demand). Moreover, a common way of pricing for Internet connection is a monthly flat rate, while other actors, e.g. APs or the so-called Over the Top Providers (OTTs) receive revenues on a per bandwidth-consumed basis, relying their services on the existent network infrastructure but not remunerating Network Providers adequately [11]. Taking into account the previous considerations many companies and academic groups are analyzing future scenarios so as to meet the end-to-end requirements and business models. As a possible architecture to provide these services, the ASs alliances or federations have emerged (see for instance [1]). In this kind of interconnection market there exists a cooperation on infrastructure, policies and incentives for rational usage of resources and agreements for providing end-to-end QoS. At the same time, interesting issues arise, such as priorities and revenue sharing. In this work, we aim at providing a framework in that sense.
We shall focus on a scenario in which ASs work together in a collaborative way in order to sell end-to-end quality assured bit pipes. The pipes are not necessarily sold to the final user but are rather sold to intermediate actors like brokers or OTT which will in turn resell them to the final user, by providing their own services through a quality assured path.
In this context, our contribution is actually twofold. On the one hand, we address the bandwidth allocation problem providing a solution through which the end-to-end quality parameters are assured and the revenue of the whole alliance is maximized. In addition, we prove that this mechanism can be carried out on a distributed fashion. On the other hand, we cover a subsequent problem that is how to distribute the revenues among all the members of the alliance. In this regard, we provide a mechanism that has fairness properties and provides incentives to the ASs to increase their features towards the federation. Beyond the specific contributions, the proposed framework links the revenue income mechanism with the revenue sharing one, which to the best of our knowledge has not been proposed in this context before.

2 Bandwidth allocation with end-to-end QoS Constraints

We are interested, as aforementioned, on a scenario where several ASs work together to sell capacity on a multidomain quality assured path. We shall refer to the quality assured path as *QoS pipe* or as *path*.
In this scenario, the capacity dedicated by each AS to sell by this means is a portion of their already deployed capacity. That is to say, ASs have their infrastructure through which traditional services are sold following the best-effort paradigm and they decide to dedicate some portion of their capacity to the federation.

For each QoS pipe there is a group of users or buyers interested on getting a portion of bandwidth on that pipe. The amount of money this group is willing to pay for each value of bandwidth is the so-called utility function. The objective is to sell the available resources in such a way that the revenue of the whole alliance is maximized while the end-to-end constraints are accomplished. We shall work with the end-to-end delay.
Let us introduce some notations so as to formally represent the scenario described above. Each AS in the alliance is abstracted to a node indexed by n with an equivalent capacity of c_n. The complete set of nodes is denoted by N. More complex topology abstrction models could be used instead of the single node aggregation. Though we leave out of the scope of this work the study of the AS topology abstraction, other alternatives and more comments can be seen in [3]. The available pipes are the ones in the set S and are indexed by s. The constraint on the delay on path s (i.e. the maximum admissible delay) is denoted by D_s. We assume that the routes within the alliance are fixed and single-path. We represent this routes with the $|N| \times |S|$ matrix R, where the notation $|\cdot|$ refers to the cardinal of the set. The entry $R_{i,j}$ is equal to 1 if the route of the pipe j traverses the node i and is equal to zero otherwise. We denote pipe's s route as $r(s)$. The bandwidth allocated to pipe s (i.e. the amount of traffic sold to the buyers associated to path s) is denoted by a_s. The utility function associated to each path s is called $U_s(a_s)$. We assume that $U_s(a_s)$ is known and, as usual in this context, it is a strictly concave function of the bandwidth.
Please note that the QoS pipes are defined by an ingress and egress point along with a maximum delay. This implies that two QoS pipes are considered different even if they share exactly the same physical path but provide different delay bounds.
Let us now state some additional assumptions. The delay introduced by each node in a path is an increasing convex function of the bandwidth carried by all the paths traversing the node. We assume that this function can be somehow learned or estimated by the domain, and we leave out of the scope of this paper the means for computing it. The delay function of node n is denoted as $f_n(a)$ where $a = \{a_s\}_{s \in S}$.
The amount of traffic sold to all paths must be such that the revenue perceived by the alliance is maximized while the QoS constraints are fulfilled. This is formalized in the following bandwidth allocation problem:

Problem 1.

$$\max_{a_s} \sum_{s:s \in S} U_s(a_s)$$
$$\text{s.t.} \sum_{n:n \in r(s)} f_n(a) \leq D_s, \forall s \in S.$$

Remark 1. In Prob. 1 we have not included a capacity constraint which is assumed to be taken into account in f_n. Indeed, if f_n is a barrier function (i.e. it approaches infinity as the bandwidth approaches the capacity) we can safely ignore any capacity constraint.

Remark 2. The fact that the association may not want to sell bandwidth on a certain path if the incomes perceived by doing so are lower than

a certain bound is not considered either. However, we can model this situation by defining a cost function of the allocated bandwidth $\kappa_s(a_s)$ for each service $s \in S$ and modifying the objective function in Prob. 1 by $\sum_{s \in S} [U_s(a_s) - \kappa_s(a_s)]$. Provided the cost function is convex, the new problem would be analogous to Prob. 1. For the sake of notations simplicity we shall not consider the cost function hereafter.

We aim at solving Prob. 1 in a distributed way. Hence, we shall explore a primal-dual approach for Prob. 1, whose associated Lagrangian is:

$$L(a, \lambda) = \sum_{s:s \in S} \left[U_s(a_s) + \lambda_s \cdot \left(D_s - \sum_{n:n \in r(s)} f_n(a) \right) \right], \qquad (1)$$

where $\lambda = \{\lambda_s\}_{s \in S}$ is the vector of Lagrange multipliers.
To find a saddle point of (1) (i.e. the optimum of Prob. 1) we use the gradient-projection algorithm updating the primal and dual variables as follows:

$$a_s^{t+1} = \left[a_s + \gamma_s \left(U_s^{'}(a_s) - \sum_{n:n \in r(s)} \sum_{v:n \in r(v)} \lambda_v f_n^{'}(a) \right) \right]^+ \qquad (2)$$

$$\lambda_s^{t+1} = \left[\lambda_s - \alpha_s \left(D_s - \sum_{n:n \in r(s)} f_n(a) \right) \right]^+ \qquad (3)$$

where $[\cdot]^+ = \max\{0, \cdot\}$ and α_s, γ_s are step sizes.
The updates (2,3) are performed iteratively on each edge router of a pipe, which we call the *source*. Every source sends an initial value for λ_s and a_s through route $r(s)$. Each node receives all the values and computes the delay, the derivative of the delay times the sum of the lambdas it has received and sends them to the source. All these values can be accumulated in two sums in the way back to the source, thus only two values are needed to be sent back to the source on each iteration. Once the source receives such values it proceeds to update the value in λ_s and in a_s. This is repeated iteratively in the control plane and it is run prior to any resource allocation.
The following theorem proves the convergence of the algorithm.

Theorem 1. *Convergence of the primal-dual algorithm. Given the Prob. 1 let $\sum_s U_s(a_s)$ be a strictly concave function and $f_n(a)\, \forall n \in N$ convex functions. Then the iterations $a_s^t\, \forall s \in S$ as defined in* (2) *and* (3) *converge asymptotically to the solution of Prob. 1.*

The proof is not provided here for lack of space reasons. Please refer to [3] for more details.

2.1 Application: Multidomain Network Auctions

We now discuss an example that fits to the model proposed before. We associate to each pipe a service to be sold which has a certain bandwidth

σ_s and an assured delay D_s (for instance, this service can be a VoD movie). Several instances of a service are sold through the same pipe.
These services are sold by means of network bandwidth auctions. In particular, we shall follow the first price auctions model where the winner user is charged with the amount he/she bids. This bidding mechanism is the most suitable to our problem as explained in Sect. 5.
We shall first consider the case of one-shot bandwidth auctions. That is to say, that the whole capacity available for providing the services is going to be auctioned at a certain moment.
Let us introduce some new notations. For each service s there are N_s buyers or users, which participate in the auction for obtaining an instance of the service. Each of the N_s users bids $b_s^{(i)}$ which we order as

$$b_s^{(1)} \geq b_s^{(2)} \geq \cdots \geq b_s^{(N_s)}. \tag{4}$$

The resource allocation decision is to find which of these bids to accept, so as to maximize the profit of the whole alliance while the per-route delay remains smaller than a given bound, under a first-price auction. Since for each s all bids are for the same bandwidth and delay constraint, the optimal solution is accepting the highest bids per service. We define the variable $\psi_{s,i}$ which is equal to 1 if bid i for service s is accepted, and zero otherwise. Then, defining the variable m_s as the number of bids accepted for service s we have the following equality:

$$\sum_{i=1}^{N_s} b_s^{(i)} \psi_{s,i} = \sum_{i=1}^{m_s} b_s^{(i)}. \tag{5}$$

Accepting m_s bids would render a total accepted rate of a_s where $a_s = \sigma_s m_s$. Thus, the utility per service can be defined as a function of a_s as

$$U_s(a_s) = \sum_{i=1}^{a_s/\sigma_s} b_s^{(i)}. \tag{6}$$

Equation (6) is defined for discrete values of a_s (the multiples of σ_s). We extend it to a piecewise linear concave function of a_s by linear interpolation.
Altogether, we can write the optimization problem as follows:

Problem 2.

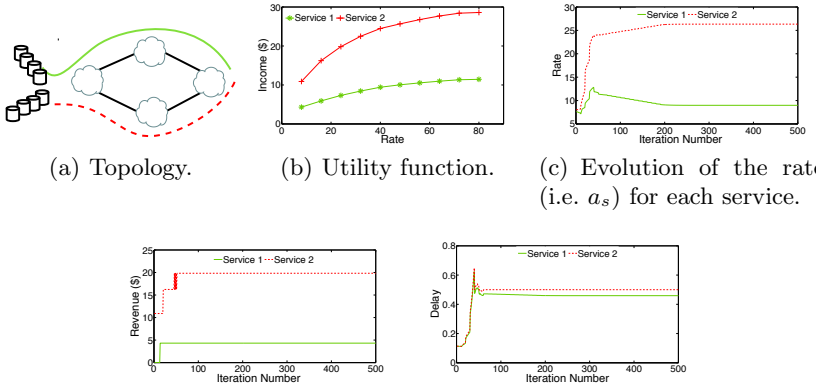
$$\max_{a_s} \sum_{s \in S} U_s(a_s)$$
$$\text{s.t.} \sum_{n: n \in r(s)} f_n(a) \leq D_s, \, \forall s \in S, \, a_s/\sigma_s \in \mathbb{Z}.$$

In Prob. 2 the objective function is concave but not strictly concave (as in Prob. 1) and an integer restriction has been added. Since integer programming is NP hard, we have strong indication of the difficulty of this problem, not easy to overcome even allowing for centralized computation. We will thus accept a sub-optimal allocation which involves solving the convex relaxation, and rounding off to satisfy the integer constraints.

The not strictly concaveness of the utility function may compromise the convergence of the algorithm by producing, in some cases, a hopping result between two consecutive integer values. In order to avoid oscillations we shall use, as proposed in [10], the so-called proximal optimization method which implies modifying the optimization problem by an equivalent one so as to have an strictly concave function as objective without changing the point at which the solution is attained. For lack of space reasons we do not provide further detail on such method.
For selling the services we repeat the process described above in a periodic fashion. Every period of time T, bids are collected and bandwidth is allocated. Most previous work on multi-period auctions (e.g. [8]) allow future bidders to compete with incumbent ones, albeit given the latter some advantage [17]. A different approach (e.g. [4]) is to impose the condition that once bandwidth has been allocated in an auction, the successful bidder has a reservation for the duration of his/her connection. Is out of the scope of this paper the specific solution for multi-period auctions problem and any of the previous proposals can be adopted.

2.2 Simulations

We present an illustrative example of the one-shot allocation mechanism. Consider the Fig. 1(a), where four ASs associate to provide two services. The equivalent capacities of all the ASs are equal to 40. Service 1 (plain path) has a delay bound $D_1 = 2$ while service 2 (dashed path) provides a delay bound $D_2 = 0.5$. Both services offer an amount of bandwidth of 8. All values are expressed in a certain coherent unit. For both services 10 buyers offer their bids. In Fig. 1(b) the resulting utility function for each service is shown. In this case the service with the most constrictive delay bound has received higher bids. Figure 1(c) shows the evolution



(a) Topology. (b) Utility function. (c) Evolution of the rate (i.e. a_s) for each service.

(d) Evolution of the revenue. (e) Evolution of the delay for each service.

Fig. 1. Bandwidth auctions with QoS constraints, one-shot allocation. Simulations.

of the rate for each service throughout the iterations needed for the convergence of the distributed algorithm. Results show that the service which implies more incomes is the one that gets more rate accepted. Figure 1(e) shows the evolution of the delays and that both constraints are accomplished. Finally, Fig. 1(d) shows the evolution of the revenue perceived per service.

3 Revenue Sharing

As we have claimed in Sect. 1, traditional peer paying in the Internet may not be suitable for these kind of assured quality services. We aim at performing the revenue sharing based on some fairness principles. Idelay, the revenue perceived by each AS should be proportional to the profit it provides to the federation. Moreover, the AS who is responsible for the end-to-end QoS degradation or bottleneck, or somehow limits incomes, should be encouraged to increase the resources dedicated to the alliance. In the following subsection we shall explore the concepts of coalitional game theory as a means of achieving the objectives mentioned above.

3.1 The Shapley Value

The Shapley value, proposed by Lloyd Shapley in 1952 [19], provides a means for performing the revenue sharing of an association or coalition. It has been widely used in the literature for its good properties. We now briefly recall some related concepts. The interested reader is referred to [20] for a complete review on Coalitional Game Theory.
A Coalitional Game with Transferable Utility is a pair (M, v) where M is a finite set of players and $v : 2^M \rightarrow \mathbb{R}$, the worth function which associates with each coalition $Q \subseteq M$ a real-valued payoff $v(Q)$ that the members can distribute among them.
Given a game $G = (M, v)$, we shall call $x = \{x_i\}_{\forall i \in M}$ to the payoff vector, where x_i represents the share of the grand coalition's (i.e. M) payoff that player $i \in M$ receives. A Pre-imputation is the set of payoff vectors such that the sum of all x_i is equal to $v(M)$. A Dummy player is a player whose contribution to the coalition is the same as the one he/she would achieve on his/her own. With these definitions the axioms of Symmetry (for any v if i and j contribute the same to any coalition then $x_i = x_j$), Dummy player (for any v if i is a dummy player then $x_i = v(\{i\})$) and Additivity are introduced, and the Shapley value is defined as follows.

Theorem 2. *Shapley Value. Given a coalitional game (M, v) there is a unique pre-imputation $\phi(M, v)$ that satisfies the symmetry, dummy player and additivity axioms and it is called the Shapley Value. It is defined, for player i as:*

$$\phi_i(M, v) = \frac{1}{|M|!} \sum_{Q \subseteq M \setminus \{i\}} |Q|!(|M| - |Q| - 1)! \left[v(Q \cup \{i\}) - v(Q)\right].$$

In addition to the properties stated on Theorem 2, the Shapley Value is efficient (it shares the total revenue) and fai. Fairness is defined in terms that for any two players $i, j \in M$, i's contribution to j is equal to j's contribution to i, that is $\phi_i(M, v) - \phi_i(M \setminus \{j\}, v) = \phi_j(M, v) - \phi_j(M \setminus \{i\}, v)$. We shall explore in the following subsection if it incentives the AS to provide better resources towards the association.

3.2 Combining the Shapley Value and the Mean Utility

In order to share the incomes perceived by means of the mechanism introduced in Sect. 2 we propose to manage two time scales. One timescale, say hourly, in which the bandwidth allocation is performed and revenue is collected. A long one, say monthly, in which the collected revenue is shared among all the ASs of the alliance. This allows for adapting the mechanism to a dynamic approach in which allocations are performed online and decentralized, and a centralized stage in which the revenue sharing is computed offline.
We define a game where the players are the set N of ASs in the association and the worth function is defined as follows. We introduce the assumption that the bids are drawn independently from a continuous probability distribution for each service. Provided this, we can safely represent the utility function of several auctions occurred during a certain period of time by the mean of all the utilities of that period. Thus, we define

$$\overline{U}_s(a_s) = E[U_s(a_s)], \tag{7}$$

which is still a strictly concave function of a_s in the general case, or a piecewise linear concave function in the case introduced in Subsect. 2.1. In addition, we assume that the delay function of every AS (i.e. f_n) remains unchanged during the considered time period.
Finally, the worth function v is defined for each sub-coalition $Q \subseteq N$ as the solution to Prob. 3, defined as:

Problem 3.

$$\max_{a_s} \sum_{s \in S^Q} \overline{U}_s(a_s)$$
$$\text{s.t.} \sum_{n:n \in r(s)} f_n(a) \leq D_s, \forall s \in S^Q,$$

where $S^Q \subseteq S$ is the set of services that can be provided by Q.
Once the revenue is collected, during several phases of bandwidth allocation, it is shared among all ASs proportional to the Shapley value. That is to say, we compute $v(Q) \forall Q \subseteq N$ according to Prob. 3 and with these values we compute the Shapley value $\phi_n \forall n \in N$. Finally, node's i revenue is computed as $\Phi_i = \phi_i \times V / \sum_{j \in n}(\phi_j)$, where V is the total revenue perceived by the coalition on the considered period
We claim that the proposed mechanism provides incentives for the ASs in the association to improve their features towards it. The features we are interested in are the ones that constitute constraints to the incomes (i.e. to Prob. 1). These features are thus captured in the node's delay

function (i.e. f_n) and we refer to them as an equivalent capacity for each AS. In the remainder of this section we formalize this property.

Theorem 3. *Incentive for improving capacities. Let (N, v, c) be a coalitional game where the set of nodes N are the players, c represents the equivalent capacities of the nodes in N and v is the worth function defined by Prob. 3. If $i \in N$ increases its capacity then its sharing coefficient (i.e. ϕ_i) will be not decreased. That is, letting c^* represent the capacities of the nodes where i's capacity is increased, $\phi_i(N, v, c*) \geq \phi_i(N, v, c)$, where $\phi_i(N, v, c)$ is the Shapley value of node i given the game (N, v) and the capacities c.*

Proof. By definition of Shapley value $\phi_i(N, v, c*) = \frac{K}{N!} \sum_{Q \subseteq N \setminus \{i\}} [v(Q \cup \{i\}, c^*) - v(Q, c^*)]$, where K is a constant and $v(Q, c^*)$ represents the worth function for subcoalition Q when the capacities are given by c^*.

$$\phi_i(N, v, c*) = \frac{K}{N!} \sum_{Q \subseteq N \setminus \{i\}} [v(Q \cup \{i\}, c^*) - v(Q, c)],$$

holds since the worth function of any coalition without i is the same, regardless the capacity of i. By subtracting i's share coefficient with and without increasing its capacity we have:

$$\phi_i(N, v, c*) - \phi_i(N, v, c) = \frac{K}{N!} \sum_{Q \subseteq N \setminus \{i\}} [v(Q \cup \{i\}, c^*) - v(Q \cup \{i\}, c)].$$

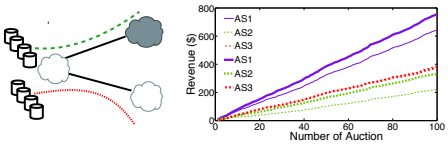
We now determine if the inequality $v(Q \cup \{i\}, c^*) \geq v(Q \cup \{i\}, c)\ \forall Q \subseteq N$ holds. Indeed, v is the solution to Prob. 3 which is the maximization of a concave function with convex constraints. By increasing the capacity we relax such problem, thus doing so yields to greater or equal solutions. □

Theorem 3 proves that if node i increases its capacity its sharing coefficient increases as well or remains the same. It is now left to be proved that the total revenue perceived by the federation in the considered period (i.e. V) does not decrease either (recall $\Phi_i = \phi_i \times V / \sum_{j \in n}(\phi_j)$). Indeed, if node's i capacity is increased either the association can allocate more bandwidth (and revenue increases) either it can allocate the same amount of bandwidth (and revenue remains the same). An argument similar to the one used before can be used to formalize this reasoning, but now considering Prob. 1 instead of Prob. 3.

3.3 Simulations

We illustrate the proposed method via a simulation with a simple example. Consider the topology shown in Fig. 2(a), where the capacities of the three ASs and their delay functions are the same. Buyers' bids are random. Results of the accumulated revenue for each AS can be seen in Fig. 2(b) represented with thin lines.

In order to explore the influence of the available capacity on the revenue sharing, we consider the topology in Fig. 2(a) but now the equivalent capacity of the shaded node is increased. The cumulative revenue sharing is shown for each AS in Fig. 2(b) in thick lines, we can see that the revenue of the AS that increases its capacity perceives an improvement.



(a) Topology. (b) Cumulative revenue sharing.

Fig. 2. Incentive for increasing capacity towards the federation. Simulation.

4 Implementation Considerations

The multidomain scenario poses new problems that are not experienced in the context of the intradomain one. For instance, political aspects (confidentiality, trust), technical aspects (interoperability, scalability) and economical ones (revenue sharing). We now briefly comment on them.
In the AS Federation context, it is usually considered that the ASs tell the truth and fulfill their common interests. Nevertheless, the ASs in the federations may ask for confidentiality, privacy on committed agreements and freedom on pricing [16].
In the distributed stage, the delay of traversing the AS and its derivative are passed from one AS to another. In the centralized stage, all the ASs in the federation send their delay function and the mean utility function to a centralized trusted entity. Thus, this framework preserves confidentiality.
Pricing can be freely defined at the per service level for the premium services, and at a per AS level for best-effort traffic.
Finally, the proposed solution appears to scale well. For the rate allocation, a few bytes in the forward and backward direction are needed during a preallocation iteration phase. For the revenue sharing, the ASs need to send reduced information to the centralized entity. The computation of the Shapley value is often #P-complete [6]. However, in our working context, the associations would rarely consist of more than ten ASs. For instance, the average AS path in the Internet is of four ASs [7]. In addition, the computation is proposed to be performed offline.

5 Related Work

The topics discussed on this paper are covered in several articles, of which we shall mention only a few of them.
Several works in the literature have proposed bandwidth network auctions for solving the bandwidth allocation problem. Most of them seek bids' truth reveling mechanisms. For instance, the ones based on Vickrey's second price auctions (e.g. [5]) where the winning user is charged the second highest bid, or the ones based on Vickrey-Clark-Groves (VCG) mechanisms (e.g. [5]) as in [8, 9, 12, 17]. Most of these mechanisms need for centralized computation, some of them assume certain network topology while others assume the buyer knows the network topology. In these cases the objective is welfare maximization. Other proposals (e.g. [4])

work with first price auctions. In this kind of auction, revenue maximization is sought and the implementation complexity is much lower than the one present in second price auctions. Moreover, in [13] it is shown that VCG mechanisms can hardly be applied on multidomain networks.
For the reasons exposed above, our auctions proposal is aligned with the one in [4]. However, we consider a multidomain federation scenario rather than a single domain and we incorporate an end-to-end QoS constraint rather than only considering capacity constraints. With respect to this last aspect [18] states a similar problem, but its context and the way it is solved differ significantly from ours.
Regarding revenue sharing, for instance, in [11] the proposal is to change the Internet economics by business contracts whose payment is determined by the Shapley value. In [21, 14] the aim is to optimize routing within an alliance of ASs and revenue is shared by means of Shapley value. We share with them the choice of using the Shapley value. However, our proposal incorporates the sell of premium services which are the sources of the revenue, and links the Shapley value with it. In addition, our approach also takes into account the features the ASs provide to the alliance rather that only considering the routing.

6 Conclusion and Future Work

We have proposed a framework for covering the complete cycle for selling end-to-end quality assured services in the context of AS federations. We have stated the problem of network bandwidth allocation with QoS constraints and showed a distributed solution. An application based on network bandwidth auctions for using such problem as the means for selling quality assured paths was shown. A mechanism for performing the revenue sharing of the federation, based on the Shapley value and the mean utility function was proposed. Such mechanism has fairness properties and was proven to incentivize ASs to increase its capacities. The behavior of the whole solution was studied through simulations.
In future work we shall enhance the interdomain network model and deepen on the delay function. In addition, we shall continue the research on revenue sharing, seeking for more properties such as the ones involving the stability of the federations and incentives to collaborate.

Acknowledgment This work was funded by the ETICS project (EC FP7 248567), cf. `www.ict-etics.eu` and the Uruguayan Agency for Research and Innovation (ANII) (PR-POS-2008-003 and FCE 2158).

References

1. ETICS: Economics and Technologies for Inter-carrier Services. European research project, supported by the 7th Framework Programme of the European Union., `http://www.ict-etics.eu`
2. Cisco Systems: Hyperconnectivity and the Approaching Zettabyte Era. Tech. rep. (Jun 2010)

3. Amigo, I., Belzarena, P., Larroca, F., Vaton, S.: Network Bandwidth allocation with end-to-end QoS constraints and Revenue Sharing in Multidomain Federations. Internal Report. Tech. rep., IIE, Universidad de la República (2011)
4. Belzarena, P., Ferragut, A., Paganini, F.: Bandwidth Allocation via Distributed Auctions with Time Reservations. In: IEEE INFOCOM, Procedings. Rio de Janeiro, Brazil, 2009 (2009)
5. Courcoubetis, C., Weber, R.: Pricing and Communications Networks. John Wiley & Sons, Ltd (2003)
6. Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. Math. Oper. Res. 19, 257–266 (May 1994)
7. Dhamdhere, A., Dovrolis, C.: Ten years in the evolution of the internet ecosystem. In: Proceedings of the 8th ACM SIGCOMM. pp. 183–196. ACM (2008)
8. Dramitinos, M., Stamoulis, G.D., Courcoubetis, C.: An auction mechanism for allocating the bandwidth of networks to their users. Comput. Netw. 51, 4979–4996 (Dec 2007)
9. Lazar, A.A., Semret, N.: Design and Analysis of the Progressive Second Price Auction for Network Bandwidth Sharing (1999)
10. Lin, X., Shroff, N.B.: Utility Maximization for Communication Networks With Multipath Routing. IEEE Transactions on Automatic Control 51(5), 766–781 (May 2006)
11. Ma, R.T.B., Chiu, D.M., Lui, J.C.S., Misra, V., Rubenstein, D.: Internet Economics: The Use of Shapley Value for ISP Settlement. IEEE/ACM Transactions on Networking 18(3), 775–787 (Jun 2010)
12. Maillé, P., Tuffin, B.: Pricing the internet with multibid auctions. IEEE/ACM Trans. Netw. 14, 992–1004 (Oct 2006)
13. Maillé, P., Tuffin, B.: Why VCG auctions can hardly be applied to the pricing of inter-domain and ad hoc networks. In: 3rd EuroNGI conference, Trondheim, Norway. pp. 36 – 39 (2007)
14. Mycek, M., Secci, S., Pioro, M., Rougier, J.L., Tomaszewski, A., Pattavina, A.: Cooperative multi-provider routing optimization and income distribution. In: DRCN 2009. pp. 281–288 (Oct 2009)
15. N. Le Sauze et al.: ETICS : QoS-enabled interconnection for Future Internet services. In: Future Network and Mobile Summit (2010)
16. Pouyllau, H., Douville, R.: End-to-end QoS negotiation in network federations. In: IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps). pp. 173–176 (2010)
17. Reichl, P., Bessler, S., Stiller, B.: Second-chance auctions for multimedia session pricing. In: PROC. MIPS03 (2003)
18. Saad, M., Leon-garcia, A., Yu, W.: Rate Allocation under Network End-to-End Quality-of-Service Requirements. In: GLOBECOM 2006
19. Shapley, L.: A value for n-person games. In: Kuhn, H., Tucker, A. (eds.) Contributions to the Theory of Games
20. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge, UK (2009)
21. Stefano Secci, Jean-louis Rougier et al.: Connection-oriented Service Management in Provider Alliances: a Shapley Value Perspective. In: EuroNF 5th Int. Workshop on Traffic Management and Engineering for the Future Internet (2009)

On The Problem of Revenue Sharing in Multi-domain Federations*

Isabel Amigo[1,2], Pablo Belzarena[1], and Sandrine Vaton[2]

[1] Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
Julio Herrera y Reissig 565 CP 11300
[2] TELECOM Bretagne, Brest, France
Technopole Brest Iroise CS 83818 29285 Brest Cedex 3
iamigo@fing.edu.uy, belza@fing.edu.uy, sandrine.vaton@telecom-bretagne.eu

Abstract. Autonomous System alliances or federations are envisioned to emerge in the near future as a means of selling end-to-end quality assured services through interdomain networks. This collaborative paradigm mainly responds to the ever increasing Internet traffic volumes that requires assured quality, and constitutes a new business opportunity for Network Service Providers (NSPs). However, current Internet business rules are not likely to satisfy all involved partners in this emerging scenario. How the revenue is shared among NSPs must be agreed in advance, and should enforce economical incentives to join an alliance and remain in it, so that the alliance remains stable. In this paper, we work on the scenario of such federations, where service selling is formulated as a Network Utility Maximization (NUM) problem. In this context, we formally formulate the properties the revenue sharing (RS) method should fulfill and argue why the existing methods are not suitable. Finally, we propose a family of solutions to the RS problem such that the economical stability and efficiency of the alliance in the long term is guaranteed. The proposed method is based on solving a series of Optimization Problems and considering statistics on the incomes.

Keywords: Revenue sharing, Optimization, Autonomous Systems Alliances.

1 Introduction

Internet traffic consumption tendencies are evolving along two main axis. On the one hand, the continuous growth in terms of volume as well as in terms of Quality of Service (QoS) demanding applications, such as telepresence, video or gaming [3]. On the other hand, the need for QoS connectivity from end-to-end across several domains or ASs, which posses political, economical and technical issues [15]. In addition, there is a need for NSPs to find new business cases and technology for fulfilling customer needs and maximizing incomes.

* This work was funded by the ETICS project (EC FP7 248567)[1] and the Uruguayan Agency for Research and Innovation (PR-POS-2008-003)

Nowadays Internet business rules for domain interconnection (peering agreements and customer-provider agreements) may not be able to provide a sustainable economy for all actors in the value chain (i.e. Application Providers, NSPs, etc.). Indeed, non of such rules are aware of the QoS capabilities of the different domains, and peering ones are based on a traffic-symmetry premise that may no longer be valid in evolving services (for instance, on-line video, which is foreseen as one of the services that is going to grow the most [3]).
Further on, a common way of pricing for Internet connection is a monthly flat rate, while other actors like Application Content Providers or the so-called Over-The Top-Providers receive revenues on a per bandwidth-consumed basis, relying their services on the existing network infrastructure but not remunerating network providers adequately [12].
Taking into account the previous considerations many companies and academic groups are analyzing different future scenarios in order to meet the end-to-end requirements by defining inter-domain architectures and business models. As a possible architecture to provide these end-to-end QoS enabled services, the concept of ASs alliances or federations has emerged (see for instance [1]). In this kind of interconnection market scenario there exists a cooperation among ASs in infrastructure, policies and incentives for rational usage of resources and agreements for providing end-to-end QoS. While at the same time, challenging issues arise, such as priorities and revenue sharing.
This work focuses on the previously described context: several ASs that create a federation and offer for sell different services. In particular, we shall work on the framework presented on [4]. In such framework, for each service there is a group of users interested in buying it. The total income is a function of the whole bandwidth demanded for all the instances of all the services. The federation may not be able to sell all the demand of bandwidth for each service, because of capacity constraints in their data networks. The objective of the federation is thus, to allocate bandwidth in such a way that the revenue of the federation is maximized.
Once the services are sold, the income has to be split among all the providers involved in the federation. The problem of how to make this sharing is not an easy one. There are some properties that must be fulfilled by the solution in order to make the sharing fair, and interesting for all the providers, such that the alliance remains stable.
This work aims at shading light into the RS problem. In this sense we provide the following contributions: formal representation of the problem and discussion of the desired properties, evaluation of existing methods which concludes that none of them are suitable for our problem, guidelines for a new method, and a solution proposal. The method is validated through simulation studies.
This paper is organized as follows. Section 2 introduces the notations used throughout the paper and states the desired properties for the RS mechanism. In Section 3 we review the most common sharing rules used in the economics field, and argue on why they are not useful for our problem. This yields to presenting in Section 4 a new method, which provides with a solution that guarantees stability and efficiency in economical terms. Simulations results that demonstrate the correct behavior of the proposed method are shown in Section 5. Finally, concluding remarks and future work are addressed in Section 6.

2 Problem Description

2.1 Definitions and Notations

We first introduce the notation needed to represent the interdomain network of providers. Because of confidentiality and scalability issues we need to abstract each provider's topology to a simpler one. In particular we shall consider each AS as a node, which is a very simple abstraction, though reasonable enough for our study. In addition, more complex topology abstraction approaches could also be applied, without implying any change in the mathematical formulation of the problem (see for instance [2],[10]). The set of ASs or nodes is called N, there are $|N|$ nodes in the network, where the notation $|\cdot|$ refers to the cardinal of the set. Each node $n \in N$ has an equivalent capacity associated to it, which we call c_n, $c = \{c_n\}_{n \in N}$ is the vector of nodes capacities. S is the set of services offered by the network, there are $|S|$ services. A service can be abstracted to a tunnel that carries bandwidth from one ingress to one egress node of the network. The routing matrix R indicates the routes of all services in S, i.e. the nodes traversed by each service $s \in S$. More formally, R is a $|N| \times |S|$ matrix whose entries $\{R_{n,s}\}_{n \in N, s \in S}$ are equal to 1 if and only if route s traverses node n, and are zero otherwise.
The amount of bandwidth traversing each service route is indicated by each component a_s, $s \in S$ of column vector $a \in R^{|S|}$. There is a utility function associated to each service $s \in S$ which is called U_s and it is a function of a_s. The utility U_s is the willingness to pay of the group of users interested in service $s \in S$. We assume that $U_s(a_s)$ is known and, as usual in this context, it is a strictly non-decreasing concave function of the bandwidth. For more details on the allocation of the bandwidth traversing the federation the reader is referred to [4].
We now introduce extra notation in order to represent the RS problem. The grand coalition is N, the set of all nodes in the network. The income is assessed by the revenue function $V : 2^{|N|} \to \mathbb{R}$ which associates to each subcoalition $\mathcal{Q} \subseteq N$ (i.e. subgroups of nodes) a real value $V(\mathcal{Q}, c^{\mathcal{Q}})$, where $c^{\mathcal{Q}}$ is the capacities vector restricted to subcoalition $\mathcal{Q}$, that is:

$$c_n^{\mathcal{Q}} = c_n \text{ if } n \in \mathcal{Q} \; ; \; c_n^{\mathcal{Q}} = 0 \text{ otherwise.} \tag{1}$$

In our problem, the revenue function V is given by the solution of Problem 1. This problem states that services are sold (i.e. bandwidth is allocated) such that the revenue of the coalition is maximized, while respecting the capacity constraints. It is thus formulated as a NUM problem [11].

Problem 1.

$$\max_{a} \sum_{s \in S} U_s(a_s)$$
$$\text{s.t.} Ra \leq c^{\mathcal{Q}}, a_s/\sigma_s \in \mathbb{N} \, \forall s \in S,$$

where σ_s is the amount of bandwidth provided by service s. We also accept the notation $V(\mathcal{Q})$ to indicate the total revenue of coalition $\mathcal{Q} \subseteq N$, where the capacities are implicit. This optimal revenue problem is not an easy integer program and its convex relaxation can be not exact.

However, since integer programming is NP hard, we accept sub-optimal solutions by convex relaxation that in many cases leads to tight solutions. We define the contribution v_n of node $n \in N$ to the coalition as $v_n = V(N) - V(N \setminus \{n\})$. The total revenue is shared among all the nodes in N according to the sharing function $\Phi : \mathbb{R} \rightarrow \mathbb{R}^{|N|}$, $\Phi = \Phi(N, c) = \{\phi_n(N, c)\}_{n \in N}$, where ϕ_n is the share corresponding to node $n \in N$. For convenience, we sometimes also use the shorter notation x to denote the RS vector, where $x \in \mathbb{R}^{|N|}$ is a column vector containing on each component x_n, $n \in N$, the revenue share of node n, when the values of N and c are implicit by context.

2.2 Desired properties of the RS mechanism

We shall now state the properties that a revenue mechanism for ASs alliances should fulfill. The idea that motivates all of them is that the ASs should be encouraged to remain in the coalition, which will occur if that makes sense from the economical point of view. The properties discussed below, and more, are usually discussed in cost/revenue sharing problems, with slightly different definitions (see for instance [9],[6]). We select from them the ones that we believe are of more relevance to our problem and formally define them.

Efficiency. The mechanism should distribute the whole revenue among all the ASs in the federation, that is

$$\sum_{n:n \in N} x_n = V(N). \tag{2}$$

Stability. In order to assure the sustainability of the federation, the mechanism should not provide incentives to any subcoalition to break the grand coalition. That is to say, no subcoalition should have economic incentives to form a smaller coalition outside the grand coalition, since this would lead to instabilities in the federation. This can be written as:

$$\sum_{n:n \in \mathcal{Q}} x_n \geq V(\mathcal{Q}), \forall \mathcal{Q} \subseteq N. \tag{3}$$

Please note that this definition also implies another interesting property usually known as *stand alone*. This means that the revenue perceived by every node $i \in N$ in the coalition is not less than the revenue it could achieve alone, i.e. $x_i \geq V(\{i\})$, $\forall i \in N$. The set of points that verify (3) constitutes the so-called *core set* in the context of Coalitional Game Theory. The reader is referred to [17] for more details on the core concept and coalitional game theory.

Monotonicity. The mechanism should provide the right incentives to the nodes to increase their resources towards the coalition. In our model, these resources are considered in the capacity. We formally define this property as follows. Given c and $\hat{c}$ two vectors of nodes capacities, such that $\hat{c}_n = c_n \; \forall n \in N \setminus \{i\}$ and $\hat{c}_i \geq c_i$ then $\phi_i(N, \hat{c}) \geq \phi_i(N, c)$.
Thus, the Monotonicity property means that if an AS increases its capacity then its revenue will as well increase or remain the same.

Fairness. We want the mechanism to be fair in the sharing. There is not a general consensus in the literature regarding the notion of fairness. However, we propose the following intuitive rules to be fulfilled. If $v_i \geq v_j$ then $x_i \geq x_j$, which is usually known as the *order preserving* property and if $v_i = 0$ then $x_i = 0$. We may also accept a weaker notion of fairness, which only asks for $x_i = x_j$ if $v_i = v_j$.

3 Existing Techniques

In this section we present existing RS techniques, which are widely used in the field of economics. A detailed review can be found in [9]. We also comment on why these techniques are not suitable for our problem.

The Proportional Share: One of the simplest way to perform the RS is the one that is proportional to the contribution of each node. Using the definitions introduced in Section 2 we write the proportional share as:

$$x_i^{prop} = \frac{v_i}{\sum_{n \in N} v_n} V(N). \tag{4}$$

The proportional share a priori seems to be a very attractive distribution rule. It fulfills the properties of Efficiency and Fairness and it is very simple to compute. However, it has the drawback that it does not always guarantee neither Stability nor Monotonicity.

The Shapley Value. The Shapley value, proposed by Lloyd Shapley in 1952 [16], provides a means for performing the RS of an association or coalition. It has been widely used in the literature for its good properties. Given a coalitional game, i.e. a pair (N, V) where N is a finite set of players and $V : 2^{|N|} \to \mathbb{R}$ any worth or revenue function, the Shapley value for player $i \in N$ is defined as:

$$x_i^{sh} = \frac{1}{|N|!} \sum_{\mathcal{Q} \subseteq N \setminus \{i\}} |\mathcal{Q}|!(|N| - |\mathcal{Q}| - 1)! \left[V(\mathcal{Q} \cup \{i\}) - V(\mathcal{Q})\right]. \tag{5}$$

Among its properties, the Shapley vale has the Efficieny, Monotonicity and a particular case of the Fairness as defined in Subsection 2.2. It also fulfills its own definition of fairness in terms that for any players $i, j \in N$, i's contribution to j is equal to j's contribution to i.
However, the Shapley value does not always provide stable solutions. That is the reason why it is not suitable for our problem. Nonetheless, its great popularity in previous work is due to the fact that it is proven that it provides with stable solutions when the revenue function is a convex function. For instance, the Shapley value has been used in [12], where the proposal is to change the Internet economics by business contracts whose payment is determined by the Shapley value. And also in [14], where the aim is to optimize the routing within an alliance of ASs and the revenue is shared by means of Shapley value. Recently, it has also been use in [18] for splitting cost savings among several domains.
In our problem, the revenue function V is not a convex one and solutions through Shapley value can lie outside the *core*. A simple example can be found using the topology on Fig. 1b and the example in Section 4. In that case, the Shapley value renders $x = (1/2, 3, 3/2)$ which is outside the *core* while we shall show on that section that a non-empty *core* exists.

The Aumann-Shapley Rule. The Aumann-Shapley Rule for cost sharing [5] was introduced by Shapley and Aumann in 1974, and can be applied analogously for a RS problem. The idea of this rule is to compute the revenue share of node $i \in N$ as its average marginal revenue along a certain path going from capacity equal to 0 to c_i. More precisely, the share for node $i \in N$ according to this rule is defined as:

$$x_i^{as} = \int_0^{c_i} \partial_i V(N, \frac{t}{c_i} c) dt = c_i \int_0^1 \partial_i V(N, tc) dt, \tag{6}$$

where the notation $\partial_i V(N, c)$ means the first order derivative of V at c with respect to c_i. Please note that in Equation (6) we have used the alternative notation for V where it is explicitly mentioned its dependency on the subcoalition and the equivalent capacities.
In first place, it must be noticed that the derivative of V with respect to c_i is not defined for all values of c_i. Indeed, consider a simple topology with only one service crossing several nodes, which all have the same capacity. Let $\hat{c}$ be that capacity. If a given node i increases its capacity, the other nodes will act as bottlenecks and the revenue will not change, while if i reduces its capacity then it will itself become the bottleneck and the revenue will decrease. Hence, the derivative of V takes different values at both sides of $\hat{c}$ and is not defined at $c_i = \hat{c}$. In addition, this rule does not fulfill the Monotonicity property, this is due to the characteristics of our revenue function. Furthermore, this rule applied to our problem could even provide incentives to reduce capacity.

The Friedman-Moulin Rule. This rule was proposed by Friedman and Moulin in 1999 [8]. We introduce the operator $\wedge$, which is defined for two vectors a and $b \in \mathbb{R}^{|N|}$ as $a \wedge b = \min(a_i, b_i)$ $i \in N$ and column vector e, which is of dimension $|N|$ and has all its components equal to one. This rule is similar to the Aumman-Shapley one, in terms that it integrates marginal revenues, but in this case the integration is done through a different path. According to the Friedman-Moulin rule, the share for node $i \in N$ is calculated as:

$$x_i^{fm} = \int_0^{c_i} \partial_i V(N, t \cdot e \wedge c) dt. \tag{7}$$

This rule can not be applied in our context since V is not derivable along the whole path, for the same reasons explained above.

4 The Proposed Method

Having seen that existing techniques are not suitable for our problem, we shall now propose a new method to perform the RS in our specific scenario. We focus our attention on two properties: Stability and Efficiency. Nevertheless, we shall present a flexible method which allows for including further properties on future work. We first study the set of possible solutions. For clarity sake, we consider this set of solutions in a simple scenario, which we call the one-shot scenario. In this scenario services are sold through what we call a service selling (SS) phase and RS is performed right afterwards. We shall latter on move to a multi-period scenario.

4.1 The Feasible Solutions Set

In order to have stability in the coalitions inequality (3) must hold. Let us enumerate all the possible subcoalitions $\mathcal{Q} \in N$ and index them using index $j = 1 \dots 2^{|N|}$. We rewrite inequality (3) as a linear system as:

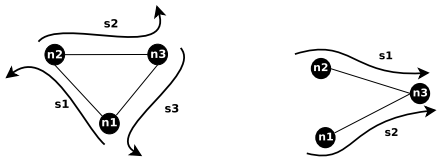
$$Qx \geq \hat{v}, \tag{8}$$

where $Q = \{Q_{j,i}\}$ is a $2^{|N|} \times |N|$ matrix that indicates which nodes belong to each subcoalition (i.e. $Q_{j,i} = 1$ if node i belongs to subcoalition j and 0 otherwise) and $\hat{v} = \{V(\mathcal{Q}_j)\}_{j=1\dots 2^{|N|}}$ is the vector that indicates in the j-th component the revenue of subcoalition j.
We must consider at the same time the Efficiency property, which we write as the vector representation of Equation (2):

$$e^T x = V(N). \tag{9}$$

But do these constraints determine a unique point? Or rather they determine a set of points? Is this set empty? We show through the following examples that actually different cases can occur.

An empty feasible set. Consider the network on Fig. 1a. The capacities of all nodes are equal to 1 unit. The three services illustrated on the mentioned figure are sold, each one of them is defined for 1 unit of bandwidth. The utility functions are such that $V(N) = V(\{1,2\}) = 5$, $V(\{2,3\}) = 4$ and $V(\{1,3\}) = 2$.



(a) (b)

Fig. 1: The feasible set in different situations.

For achieving stability the total revenue (5 units) must be split in such a way that every route receives at least what they would receive alone. It is not difficult to see that this is not possible at the same time for all routes, since the following inequalities must hold: $x_1+x_2 \geq 5$, $x_1+x_3 \geq 2$, $x_2 + x_3 \geq 4$ and $x_1 + x_2 + x_3 = 5$. Hence, the feasible set is empty.
It is interesting to remark that for different values of the revenues, and same topology, the feasible set could be non-empty.

A feasible region. Consider now the network on Fig. 1b. The capacities are again equal to 1 unit for all nodes and we sell services of 1 unit of bandwidth. Utility functions are such that $V(N) = V(\{2,3\}) = 5$ $V(\{1,3\}) = 2$. A feasible solution must fulfill $x_1 + x_3 \geq 2$, $x_2 + x_3 \geq 5$ and $x_1 + x_2 + x_3 = 5$. The vectors x that satisfy all equations are $\{x = (0, 2+\epsilon, 3-\epsilon) : \epsilon \in \mathbb{R}, 0 \leq \epsilon \leq 3\}$, which corresponds to a segment in $\mathbb{R}^2$.

4.2 One-Shot Scenario

We have seen in the previous subsection that configurations with no solution can exist, in this case we claim that the coalition should not exist as such, since there is no RS method that can make it stable. Therefore, we focus our attention on the case where constraints (8) and (9) determine a region. In order to choose a point from such region we formulate the following Optimization Problem:

Problem 2.

$$\min_{x} \ f(x)$$
$$\text{s.t. } Qx \geq \hat{v},\ e^T x = V(N),$$

where $f(x)$ is a convex function. Please note that we can dispense with the restriction of non negative revenue shares, since it is already considered on the Stability property. Problem (2) constitutes a family of methods which can be tuned to cover additional properties by considering different objective functions. Examples of objective functions are to project the Proportional share or the Shapley value into the feasible set, which means the method inherits their properties when the share is already in the feasible set, and otherwise returns the closest value. Yet another example is the square of the Euclidean norm of the share vector, which would intuitively provide with more even shares among the nodes. Regarding implementation aspects, the proposal is to have a central trusted entity computing the RS. This entity must know the utility functions for each service and the topology of the coalition, at the AS level.

4.3 Multi-period Scenario

We shall now focus our attention on the multi-period scenario, that is to say, when several phases of SS occur. A new period implies a new utility function, thus different values for $U_s, s \in S$. This necessarily leads to a different feasible set. Finding on each period a valid RS vector would involve performing a great number of computations, besides to a great exchange of information among the domains and the central entity solving the RS. In other words, the multi-period case may pose the problem of scalability thus, we face the challenge of providing a scalable approach. One could naively propose as a solution to compute the RS once, and then simply keep the sharing proportion for the subsequent RS phases. However, if we were to use the same RS proportion for a new SS instance, then the new RS vector can lie within the new feasible set or outside of it, which leads us to discard that option.
Altogether, we are motivated to perform the RS on a longer timescale than the SS phase, and work with statistics of the utilities received during the several SS phases considered for a given RS phase. In the following we shall discuss two different approaches for working with such statistics.

Approach 1. In order to model the multi-period situation, let us introduce the assumption that the utility functions depend on a value drawn independently from a continuous probability distribution for each service.

Provided this, we can safely represent the utility functions of several SS phases occurred during a certain period of time by their mean over that period of time. As usually, notation E represents the expectation of a random variable. We define the mean utility function as:

$$\overline{U}_s(a_s) = E[U_s(a_s)], \tag{10}$$

which is still a non-decreasing concave function of $a_s, \forall s \in S$. Finally, we redefine the revenue function V by Problem 3, and call it $\mathcal{V}$.

Problem 3.

$$\max_a \sum_{s \in S} \overline{U}_s(a_s)$$
$$\text{s.t.} Ra \leq c^{\mathcal{Q}}.$$

The procedure then continues solving Problem 2, but considering now $\mathcal{V}$ instead of V for the definition of $\hat{v}$.
The explained mechanism allows us to perform the computation only once in a while (e.g. monthly). In addition, the amount of information exchanged is also kept small, since the only information that has to be transmitted to the central entity on each RS phase is the mean of the utilities over that period. However, can we be sure that the solution provided by this approach is always stable in the long term? The answer to this question is addressed in the following.

Approach 2. Usually, providers' decisions are based on long term behaviors, mainly for keeping network stability. Likewise, the interest of the providers to remain in the alliance would be based on its economical stability in the long term. That is, they would likely need to know if their revenue share is economically attractive in the long term. For considering such situation we compute the long term feasible set, which is obtained after the expectation of the revenues of each subcoaliton, and obtain the RS from such set. This is summarized on Problem 4.

Problem 4.

$$\min_x \ f(x)$$
$$\text{s.t. } Qx \geq E\{\hat{v}\},\ e^T x = E\{V(N)\}.$$

Please note that $E\{\hat{v}\}$ is obtained by computing the expectation of the output of Problem (1) for each subcoalition $\mathcal{Q} \subseteq N$.
The raised question reduces then to answering if the point chosen by Problem (3) lies within the feasible set of Problem (4) or not. Unfortunately this is not necessary true. Indeed, as shown in [13], where relationships between stochastic non-liner programming problems are demonstrated, the following inequality applies:

$$E\{V(\mathcal{Q})\} \geq \mathcal{V}(\mathcal{Q}), \forall \mathcal{Q} \subseteq N, \tag{11}$$

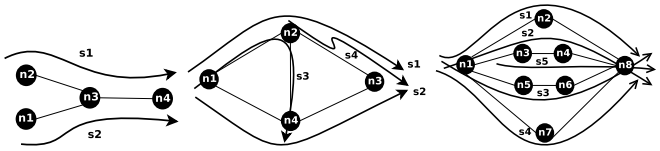
which means that the feasible set of Approach 2 is contained in the one of Approach 1. However, we have no indication about the tightness of the bound, thus we shall evaluate the impact of using either of both approaches by simulation, in the following section.

5 Simulations

The simulations presented in this section were performed on a regular computer with a i5 processor of 2.67GHz and 3.6 GB of RAM memory. The optimization problems were solved using CPLEX through AMPL.

5.1 One-shot scenario

We shall consider the topologies on Fig. 2, where the arrows indicate the services' paths, $c_n = 10$ for all nodes n and $\sigma_s = 5$ for all services s, all values expressed in some coherent unit. Tabel 1a shows the utility in \$ for carrying 5 and 10 units of bandwidth, where the underlined values correspond to the solution of Problem 1 (i.e. the revenue).



(a) 1. (b) 2. (c) 3.

Fig. 2: Topologies used throughout the simulation studies.

	Utility (\$)					
Service	$U_s(5)$	$U_s(10)$	$U_s(5)$	$U_s(10)$	$U_s(5)$	$U_s(10)$
s1	1	2	5	9	<u>7</u>	8
s2	6	<u>9</u>	<u>7</u>	11	2	3
s3	-	-	<u>11</u>	16	5	8
s4	-	-	<u>12</u>	18	5	8
s5	-	-	-	-	<u>6</u>	11
	Topology 1		Topology 2		Topology 3	

(a) Utility values

Topology	Revenue Share (%)					v_i
	x^{norm}	$x^{sv}_{\perp}$	x^{sv}	$x^{prop}_{\perp}$	x^{prop}	
1	0.333	0.284	0.278	0.280	0.280	7
	0	0	0.019	0	0	0
	0.333	0.358	0.352	0.360	0.360	9
	0.333	0.358	0.352	0.360	0.360	9
2	0.200	0.175	0.175	0.200	0.211	12
	0.300	0.353	0.353	0.344	0.333	19
	0.300	0.297	0.297	0.256	0.246	14
	0.200	0.175	0.175	0.200	0.211	12
3	0.077	0.130	0.183	0.077	0.105	2
	0.077	0.023	0.076	0.077	0.105	2
	0.039	0.039	0.106	0.039	0.053	1
	0.039	0.039	0.106	0.039	0.053	1
	0	0	0.014	0	0	0
	0	0	0.014	0	0	0
	0	0	0.036	0	0	0
	0.769	0.769	0.465	0.769	0.684	13

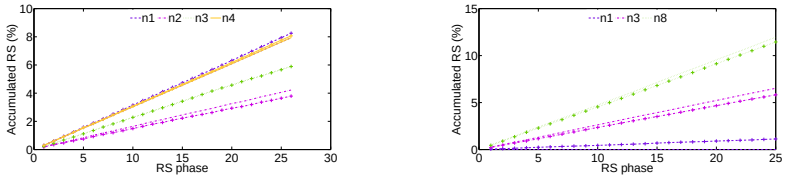
(b) Results using different criteria.

Table 1: Revenue sharing, one-shot scenario.

Revenue shares were computed using the different rules introduced in Section 3 and the criteria introduced in Section 4. Results are shown on Table 1b (where notation $x^*_{\perp}$ stands for the projection of * into the feasible set and x^{norm} is the solution when $f(x) = \|x\|^2$), along with the value of v_i for each node. Topologies 2 and 3 constitute examples where the Proportional share does not lie into the feasible region, so do topologies 1 and 3 for the case of the Shapley value. Regarding the different criteria, x^{norm} shows the most even shares. Some fairness notions are also observed for most criteria but the Shapley value; the smaller the v_i, the smaller the x_i, nodes whose contribution is null (i.e. $v_i = 0$) obtain no revenue, while those with same v_i obtain the same RS.

5.2 Multi-period scenario

We now compute the solution according to Approach 1 and 2. In both cases, a number of 50 SS phases were performed before a RS phase and the projection of x^{prop} was used as criteria. Results for topology on Fig. 2b are shown on Fig. 3a. For this topology, on every RS phase the results obtained using both approaches are almost the same. Same thing occurs for all the simulations performed, in particular for the topology on Fig. 2c, whose results for selected nodes are shown on Fig. 3b.



(a) Topology on Fig. 2b. (b) Topology on Fig. 2c.

Fig. 3: RS using Approach 1 (-) and 2 (+).

We now evaluate the time consumed by each approach. We shall consider a simple topology with only one service defined and linearly increase the number of nodes in the service's path. Results show that for both approaches the time consumed by the method increases exponentially with the number of nodes in the network. This is related to the Stability property, since for taking it into account we consider all subcoalitions of nodes (i.e. $2^{|N|}$ cases). For a topology of 8 nodes, Approach 1 consumed on the average 2 ms while Approach 2 consumed 135 ms. However, Approach 2 is still feasible, moreover considering it is proposed to be performed off-line and in a long timescale. In addition, ASs alliances are likely to have no more than 10 nodes, considering for instance, that the average AS path in the Internet is of 4 ASs [7].
All in all, we can claim that Approach 2 provides with a solution that fulfills the sought properties with affordable computation time.

6 Conclusion and Future Work

The present work has addressed the problem of RS in the context of ASs alliances. We have focused on the case where the income of the alliance is determined by the output of a NUM problem. This particular scenario poses new challenges. Previous results for performing RS, which have been reviewed in this work, were found to be inappropriate applied to this case. The desired properties for the RS were formally stated and a new method has been proposed. This method is conceived for providing economical stability and efficiency to the alliance and it is flexible enough to be adapted to fulfill additional properties. The method is based on solving optimization problems and considers statistics on the income. Its proper behavior has been evaluated through simulation studies.
As future work, we shall study the inclusion of further properties into the method, as well as research on the relation between the two approaches

provided for the multi-period scenario. In particular, we are interested in including the consideration of QoS parameters associated to the services and providing incentives through the RS mechanism to guarantee them.

References

1. ETICS: Economics and Technologies for Inter-carrier Services. European research project of the FP7, http://www.ict-etics.eu
2. The ATM Forum, Private Network-Network Interface, specification version 1.0, March 1996
3. Cisco Systems: Entering the Zettabyte Era. Tech. rep. (2011)
4. Amigo, I., Belzarena, P., Larroca, F., Vaton, S.: Network bandwidth allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations. In: ICQT 2011. pp. 50–62. Springer-Verlag, Berlin, Heidelberg (2011)
5. Aumann, R.J., Shapley, L.S.: Values of non-atomic games. Princeton University Press (1974)
6. Bogomolnaia, A., Holzman, R., Moulin, H.: Sharing the Cost of a Capacity Network. Mathematics of Operations Research 35(1), 173–192 (Feb 2010)
7. Dhamdhere, A., Dovrolis, C.: Ten years in the evolution of the internet ecosystem. In: ACM SIGCOMM 2008. pp. 183–196. ACM (2008)
8. Friedman, E., Moulin, H.: Three Methods to Share Joint Costs or Surplus. Journal of Economic Theory 87(2), 275–312 (Aug 1999)
9. Hougaard, J.L.: An Introduction to Allocation Rules. Springer (2009)
10. Karaoglu, H.T., Yuksel, M.: Value flows: Inter-domain routing over contract links pp. 327–332 (Dec 2010)
11. Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. The Journal of Operational Research Society 49(3), 237–252 (1998)
12. Ma, R.T.B., Chiu, D.M., Lui, J.C.S., Misra, V., Rubenstein, D.: Internet Economics: The Use of Shapley Value for ISP Settlement. IEEE/ACM Transactions on Networking 18(3), 775–787 (Jun 2010)
13. Mangasarian, O.L.: Nonlinear Programming Problems with Stochastic Objective Functions. Management Science 10(2) (1964)
14. Mycek, M., Secci, S., Pioro, M., Rougier, J.L., Tomaszewski, A., Pattavina, A.: Cooperative multi-provider routing optimization and income distribution. In: DRCN 2009. pp. 281–288 (Oct 2009)
15. N. Le Sauze et al.: ETICS : QoS-enabled interconnection for Future Internet services. In: Future Network and Mobile Summit (2010)
16. Shapley, L.: A value for n-person games. In: Kuhn, H., Tucker, A. (eds.) Contributions to the Theory of Games
17. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge, UK (2009)
18. Stanojevic, R., Castro, I., Gorinsky, S.: CIPT: using tuangou to reduce IP transit costs. In: CoNEXT 2011. ACM, New York, NY, USA (2011)