Actes du 11ème Atelier en Évaluation de Performances

Urtzi Ayesta, Balakrishna Prabhu, Ina Maria Maaike Verloop

Actes du 11ème Atelier en Évaluation de Performances

Toulouse, 15–17 mars 2016

Éditeurs :
Urtzi AYESTA, Balakrishna PRABHU, Ina Maria VERLOOP

site web : aep11.sciencesconf.org/

Soutiens :

Table des matières

Avant-propos

Objectif

L'Atelier en Évaluation de Performances est une réunion destinée à faire s'exprimer et se rencontrer les jeunes chercheurs (doctorants et nouveaux docteurs) dans le domaine de la Modélisation et de l'Évaluation de Performances, une discipline consacrée à l'étude et l'optimisation de systèmes dynamiques stochastiques et/ou temporisés apparaissant en Informatique, Télécommunications, Productique et Robotique entre autres.

La présentation informelle de travaux, même en cours, y est encouragée afin de renforcer les interactions entre jeunes chercheurs et préparer des soumissions de nouveaux projets scientifiques. Des exposés de synthèse sur des domaines de recherche d'actualité, donnés par des chercheurs confirmés du domaine renforcent la partie formation de l'atelier.

Historique

Démarré sous l'impulsion de l'équipe de Raymond Marie en 1991, l'atelier en évaluation de performance a eu lieu à Rennes (1991-93), Grenoble (95), Versailles (96), Paris ENS (01), Reims (03), Aussois (08) et Sophia Antipolis (14). L'objectif est de fédérer une communauté allant des probabilités aux expérimentations en informatique. Elle recouvre donc d'une part les thématiques portant sur la modélisation des systèmes complexes avec les méthodes théoriques récentes, les environnements logiciels d'évaluation de performances jusqu'aux retours d'expérimentations en vraie grandeur.

Comité Scientifique

Le comité scientifique est composé de : Jean-Michel Fourneau, Bruno Gaujal, Marc Lelarge, Jean Mairesse, Laurent Truffet, et Bruno Tuffin.

Comité d'Organisation

Le comité d'organisation de la 11ème edition est composé de : Urtzi Ayesta, Balakrishna Prabhu et Maaike Verloop. Il a pu compter sur l'aide précieuse de Caroline Malé.

Soutiens

L'Atelier a reçu le soutien du : ANR RACON, CNRS, GDR ASR, GDR RO, IRIT, LAAS, et NOKIA.

Programme

Mardi 15 mars 2016

12h00-12h30 : Accueil des participants au LAAS-CNRS

12h30-14h00 : Repas au hall central du LAAS-CNRS

14h00-15h00 : **Exposé de synthèse**

Patrick Loiseau
Strategic resource allocation in adversarial environments (p. 5)

15h00-15h15 : Pause café

15h15-16h15 : **Session : Performance Evaluation and Quality of Service, I**

Farah Ait Salaht
Bornes stochastiques sur les mesures de performance pour des réseaux de files d'attente en tandem (p. 9)

Marziyeh Bayati (cet article n'a pas pu être presenté)
Managing Energy Consumption and Performance in Data Centers (p. 11)

16h15-16h30 : Pause café

16h30-17h30 : **Session : Performance Evaluation and Quality of Service, II**

Yves Mocquard
Compter avec les Protocoles de Population (p. 13)

Jean Marie Garcia, Mohamed El Hedi Boussada
Evaluation des performances bout en bout du trafic TCP sous le régime "Équité Équilibrée" (p. 15)

18h30 - 19h30 : Réception avec cocktail à la mairie de Toulouse

Mercredi 16 mars 2016

09h30-10h30 : **Tutoriel**

Rudesindo Núñez-Queija
Asymptotic analysis techniques for performance evaluation - Part I (p. 6)

10h30-10h45 : Pause café

10h45-12h15 : **Session : Game Theory**

Stéphane Durand, Bruno Gaujal
Average complexity of the Best Response Algorithm in Potential Games (p. 17)

Baptiste Jonglez, Bruno Gaujal
Optimal adaptive routing in packet-switched networks (p. 19)

Josu Doncel, Nicolas Gast, Bruno Gaujal
A Mean-Field Game with Explicit Interactions for Epidemic Models (p. 21)

12h15-14h00 : Repas au hall central du LAAS-CNRS

14h00-15h00 : **Exposé de synthèse**

Nidhi Hegde
Tools for the analysis of content dissemination in social networks (p. 5)

15h00-15h30 : Pause café

15h30-17h00 : **Tutoriel-Démo**

Jean-Michel Fourneau, Jean-Marc Vincent et Alain Jean-Marie
Outils logiciels du projet ANR MARMOTE (p. 7)

19h30- : Dîner en ville

Jeudi 17 mars 2016

09h30-10h30 : **Tutoriel**

Rudesindo Núñez-Queija
Asymptotic analysis techniques for performance evaluation - Part II (p. 6)

10h30-10h45 : Pause café

10h45-12h15 : **Session : Control**

Maialen Larranaga, Onno J Boxma., Rudesindo Núñez-Queija, Mark S. Squillante
Efficient content delivery in the presence of impatient customers and multiple content types (p. 23)

Jérémie Leguay, Lorenzo Maggi, Moez Draief, Stefano Paris, Symeon Chouvardas
Admission Control with Machine Learning in Software Defined Networks (p. 25)

Nicolas Jara, Reinaldo Vallejos, Gerardo Rubino
Blocking Evaluation of dynamic WDM networks without wavelength conversion (p. 27)

12h15-14h00 : Repas au hall central du LAAS-CNRS

14h00-15h30 : **Session : Streaming and resource allocation**

Zakaria Ye, Rachid El Azouzi, Tania Jimenez, Stefan Valentin
Backward-Shifted Coding (BSC) for HTTP Adaptive Streaming (p. 29)

Apostolos Destounis, Georgios Paschos, Iordanis Koutsopoulos
Resource Allocation for in-Network Computations (p. 31)

Imen Triki, Rachid El Azouzi, Majed Haddad
Anticipating Resource Management and QoE Provisioning for Video Streaming (p. 33)

15h30-15h45 : Pause café

15h45-16h15 : **Démo**

Ahmad Al Sheikh
NEST - A demonstration on network modeling and simulation (p. 7)

16h15-17h00 : **Discussions autour de l'organisation du prochain atelier**

Exposés invités

Exposés de synthèse

Nidhi Hegde
Nokia, Paris

Tools for the analysis of content dissemination in social networks

Résumé : Information propagation in networks has been studied for decades. In the past, such "rumour spread" has been based on a model where a source emits an information which then spreads in the network according to push, pull, or hybrid mechanisms. The goal in such work has been to characterize the rate of spread across the whole network. More recently, information propagation from the perspective of social networks has been studied, where a more realistic model of multiple sources of information, and multiple 'types' of information is considered. In such a model, each node in the network has resource constraints and must choose on how to relay the information. In this overview, we will go over some recent work on information spread in social networks, covering analysis and algorithms for efficient dissemination. In particular, we will consider distributed algorithms for optimal information propagation, and we review game-theoretic tools used in the characterisation of such distributed control problems in this model and its variants.

Patrick Loiseau
EURECOM Institute, Sophia Antipolis

Strategic resource allocation in adversarial environments

Résumé : Allocation of resources is a well-known problem that is often solved by optimization techniques. In adversarial environments, however, the objective function (or payoff) depends on on how an adversary allocates his resources. Examples of such situations are numerous and include allocation of security defenses to different targets (where the payoff depends on how an attacker allocates his attack resources) and allocation of advertisement/lobbying resources to customers/voters (where the payoff depends on how a competitor allocates his resources). In such scenarios, the resource allocation problem becomes a game. In this talk, we review strategic resource allocation games. The fundamental model of strategic resource allocation is the Colonel Blotto game, proposed by Borel in 1921. Two players allocate an exogenously given amount of resources to a fixed number of battlefields with given values. Each battlefield is then won by the player who allocated more resources to it, and each player maximizes the aggregate value of battlefields he wins. Despite its apparent simplicity, the Colonel Blotto game is still unresolved in its most general form. We review existing solutions and briefly mention some interesting variants of this game.

Tutoriel

Rudesindo Núñez Queija
Univ. of Amsterdam et CWI, Amsterdam

Asymptotic analysis techniques for performance evaluation

Résumé : For many queueing systems exact analysis of performance measures such as queue lengths, waiting times and sojourn time is often out of reach. Also, average values may not even be the most informative measures to describe a system's performance, but one may rather be interested in performance quantiles for example. For such cases a wide range of asymptotic techniques are available that may serve to develop suitable approximations and provide valuable insights. In this course we will briefly outline several such techniques (large deviations and tail asymptotics, fluid and diffusion limits, perturbation analysis, heavy traffic limits) and illustrate them on queueing models such as GPS queues, DPS queues, and bandwidth-sharing networks. The main part of the tutorial will focus on a more detailed discussion of two particular techniques :

– Heavy-tailed asymptotics : We will explain the fundamental difference with light tailed asymptotics ("conspiracy" versus "disaster" scenarios) and illustrate several analysis techniques that one may resort to in obtaining asymptotically accurate estimates, including analytic asymptotics, probabilistic bounds and coupling arguments.
– Perturbation analysis (in particular time-scale separation) : analyzing Markovian queueing networks as multi-dimensional Markov processes may be notoriously difficult. One abstraction is to isolate the behavior of a single queue, and capture the influence of other queues in what is called the random environment. As the random environment changes state, the queue can move from one mode of operation to another (for example from lightly loaded conditions to overloaded conditions and back). Perturbation techniques provide approximations when the state changes of the random environment occur on a much faster or much smaller time scale than the queueing dynamics.

Tutoriel-Démo

Jean-Michel Fourneau[a], Jean-Marc Vincent[b], Alain Jean-Marie[c]
[a] Université de Versailles Saint Quentin
[b] Université Joseph Fourier
[c] INRIA Sophia-Antipolis Méditerranée

Outils logiciels du projet ANR MARMOTE

– Jean-Michel Fourneau : Xborne : génération, comparaison, résolution de chaînes de Markov.
– Jean-Marc Vincent : Psi3 : modélisation par événements et simulation exacte de chaînes de Markov.
– Alain Jean-Marie : marmoteCore : créer et résoudre des chaînes de Markov en C++.

Démo

Ahmad Al Sheikh
QoS Design, Toulouse

NEST - A Demonstration on Network Modeling and Simulation

Résumé : In this technical demonstration we will highlight key functionalities of NEST, QoS Design's software suites aimed at network operators. We will first demonstrate NEST IP/MPLS as we focus on defining an IP/MPLS network and associated parameters and protocols. Features such as tracing traffic flow routes throughout the network interfaces, editing access populations and technologies, and interpreting main simulation and performance evaluation results will be presented. Afterwards, we will overview other NEST software concerning optical networks and network supervision.

Bornes stochastiques sur les mesures de performance pour des réseaux de files d'attente en tandem

Farah AIT SALAHT *

LIP6, Université Paris Ouest Nanterre
4 place Jussieu, 75252 Paris cedex 05, France
Email : farah.aitsalaht@u-paris10.fr

1. Introduction

Nous nous sommes intéressés dans ce travail à l'évaluation de performance de réseaux de files d'attente en tandem, en temps discret. L'analyse numérique exacte de ces systèmes est très difficile, voire impossible à effectuer lorsque leur taille est grande. Parmi les méthodes souvent utilisées dans la littérature, on cite la simulation qui est simple et l'approche par décomposition. Cependant, ces approches ne sont que des méthodes d'analyse approximatives. Dans ce travail, nous proposons d'avoir une démarche tout à fait différente qui consiste à déterminer des bornes plutôt que des approximations sur les processus exacts de sortie du réseau (distributions de sortie, distribution de pertes et délais de bout en bout).
Pour notre étude, nous avons considéré des arrivées par batch, des services déterministes et des files d'attente à capacité finies. Nous avons ainsi développé quatre approches d'analyse distinctes permettant de déterminer des bornes stochastiques sur les mesures de performance du réseau sans aucune hypothèse sur le trafic en entrée. Pour ces systèmes, nous avons montré grâce à certains résultats théoriques prouvés [1] et sous certaines conditions, la possibilité de déterminer des encadrements fiables sur les mesures de performance. La garantie de la qualité de service représente bien évidemment notre objectif principal dans ce travail.

2. Bornes sur les mesures exactes des réseaux de files d'attente en tandem

Nous considérons un réseau de files d'attente en tandem en temps discret noté comme suit : $H_1/D/S_1/B_1 \rightarrow /D/S_2/B_2 \rightarrow \ldots \rightarrow /D/S_N/B_N$, composé de N files d'attente en série. Chaque file d'attente i possède un tampon de longueur finie B_i et un service constant de capacité S_i. Le système est supposé vide au départ. Les données entrent dans la première file d'attente selon une distribution d'arrivée H_1, puis passent à travers les files d'attente dans l'ordre : les données sortantes de la première file d'attente entrent au bout d'au moins un intervalle de temps dans la deuxième file d'attente (système slotté), et ainsi de suite (voir fig. 1).

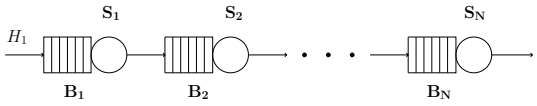

FIGURE 1 – Réseau en tandem composé de N files d'attente.

Lors de l'analyse de réseaux en tandem, nous remarquons que lorsque la séquence des capacités de service est croissante, l'analyse du réseau revient à étudier uniquement la première file. Sinon, des modifications peuvent être apportées au réseau initial simplifiant ainsi sa résolution. Pour cela nous introduisons les notions suivantes :

Definition 1 (Bottlenecks)

– *Une file* i *est bottleneck local (noté* BL*) si*
$\forall j < i, \quad S_j > S_i$.
– *Une file* i *(*$i \geq 1$*) est bottleneck global, noté* BG *si*
1. $\forall j > 1,\ j \neq i : \quad S_j > S_i$;
2. *ou si* $S_j = S_i$, *alors* $i < j$.

Selon l'emplacement des *bottlenecks* dans le réseau, nous distinguons les cas suivants :

1. ***Bottleneck global* en tête du réseau.** Le fait que la file d'entrée du réseau soit un *BG* revient à dire que les files de 2 à N ne perdent aucune données. Dans ce cas, l'analyse du système peut être effectuée de façon exacte. La première file est analysée. Pour les autres files, seuls les délais de traversée seront considérés.
2. **Sinon**, nous proposons de dériver une forme réduite du réseau initial (postprocessing sur le délai et la taille du réseau) comme suit :
   (a) Si nous avons une file i telle que $i >$*BL* (ou $i >$*BG*) et i n'est ni *BL* ni *BG*, alors la file i est supprimée du réseau.
   (b) Si nous avons une file i telle que $i > 1$ et qui est située avant le premier *BL*, alors nous pouvons également l'éliminer.

Nous définissons au final un nouveau réseau réduit, composé uniquement de la première file d'attente du réseau initial, des bottlenecks locaux et

∗. Ce travail a été réalisé avec H. Castel-Taleb, J.M. Fourneau et N. Pekergin.

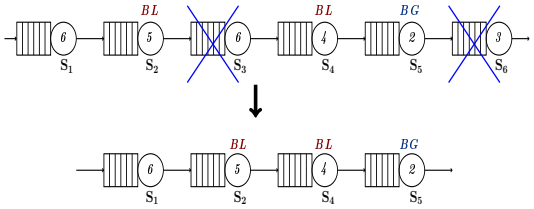

FIGURE 2 – Exemple de simplification d'un réseau en tandem.

du bottleneck global. Notre but à présent consiste à analyser ce nouveau réseau et déterminer des bornes sur ses mesures de performances. Une généralisation au réseau initial sera ensuite effectuée en ajoutant les temps de traversée des files non considérées. Nous proposons quatre approches d'analyse permettant de définir des bornes stochastiques (bornes "*st*") supérieures ou inférieures sur les paramètres du réseau en tandem : nombre de données traitées, délais de bout en bout et nombre de pertes dans le réseau. Notons que la comparaison stochastique [3] est utilisée dans ce travail. Une brève description de la nature des bornes ainsi que de la démarche suivie pour la construction de ces approches est donnée ci-après.

2.1. Approche 1 : Bornes st-supérieures sur la distribution de sortie et le délai de traversée du réseau

Nous utilisons dans cette approche le théorème de Friedman [2] sur l'interchangeabilité des files dans un réseau en tandem. Ce résultat stipule que pour des tampons de capacité infinis, le délai de bout en bout dans un réseau en tandem ne dépend pas de l'ordre des files dans le réseau. Ainsi, reposant sur ce résultat, nous proposons de fixer toutes les longueurs des tampons des files d'attente du réseau à l'infinie, ce qui nous permet de dériver des bornes supérieures sur les mesures de performance du réseau initial. Puis, en utilisant la propriété d'interchangeabilité, de déplacer la file *bottleneck* global en tête du réseau et se ramener ainsi à l'analyse d'une seule file d'attente (file *BG*).

2.2. Approche 2 : Bornes st-supérieures sur les performances d'une file i

Pour calculer les performances d'une file i du réseau, nous proposons de suivre les étapes suivantes : 1) fixer tous les tampons des files $j < i$ à l'infini, ce qui nous permet de définir des bornes supérieures sur les performances de la file i ; 2) déterminer le *BG* parmi les files 1 à $i-1$ et utiliser le théorème d'interchangeabilité de Friedman [2] pour déplacer la file *BG* en tête du réseau et enfin 3) éliminer les files 2 à $i-1$. Cette approche nous ramène à l'étude du système résiduel composé de la file 1 (file *BG*) et de la file i.

2.3. Approche 3 : Bornes st-inférieures sur les performances d'une file i

Pour cette approche, nous fixons les longueurs des tampons des files $j < i$ à zéro. En effectuant cette modification, nous définissons des bornes inférieures sur les paramètres de sortie de la file i. De plus, l'analyse du réseau peut se résumer à l'analyse de la file i uniquement, tel que l'histogramme d'entrée de la file i correspond au filtrage de la séquence d'entrée H_1 du réseau sur tous les services des files d'attente qui précèdent la file i.

2.4. Approche 4 : Bornes st-supérieures sur les performances d'une file i

Pour une file d'attente i ($i > 1$) du réseau, nous proposons de fixer les capacités de service des files $j < i$ à l'infini. Cette modification permet non seulement de dériver des bornes supérieures sur les nombres de données en sortie et le nombre total de pertes dans le réseau, mais également de simplifier son analyse en se ramenant à l'analyse de la file d'attente i uniquement avec comme séquence d'entrée H_1.

3. Conclusion

Pour l'analyse d'un réseau en tandem, nous proposons dans un premier temps une simplification du réseau initial par la recherche de bottlenecks. Nous proposons par la suite d'utiliser l'une des quatre approches proposées afin de dériver des bornes inférieures ou supérieures prouvées sur des indices de performance du réseau. Grâce à ces approches, on se ramène souvent à l'analyse de réseaux réduits plus simples qui permettent d'avoir des garanties sur les mesures de performances exactes du réseau. Notons également que cette approche peut être étendue aux réseaux en arbres, car l'analyse de ces réseaux peut s'effectuer en décomposant le réseau initial en sous-réseaux en tandem.

Bibliographie

1. F. Aït-Salaht. *Chaînes de Markov Incomplètement Spécifiées : analyse par comparaison stochastique et application à l'évaluation de performance des réseaux*. PhD thesis, Université de Versailles Saint-Quentin, 2014.
2. H. D. Friedman. *Reduction methods for tandem queueing systems* 13 : 121–131, 1965.
3. A. Müller and D. Stoyan. Comparison Methods for Stochastic Models and Risks. Wiley, New York, NY, 2002.

Managing Energy Consumption and Performance in Data Centers

M. Bayati

Affiliation
LACL
Université Paris Est Créteil
France
assal.bayati@lacl.fr

1. Introduction

The increasing development of *Date Centers* is causing problems in energy consumption. More than 1.3% of the global energy consumption is due to the electricity used by data centers, a rate that is increasing, revealed by a survey conducted in [3], which says a lot about the increasing evolution of data centers. Therefore, to ensure both a good performance of services offered by these data centers and reasonable energy consumption, a detailed analysis of the behavior of these systems is essential for designing efficient optimization algorithms to reduce the energy consumption. Two requirements are in conflict : (i) Switching on a maximum number of servers leads to less waiting time and decreases the loss of jobs but requires a high energy consumption. (ii) Switching on a minimum number of servers leads to less energy consumption, but causes more waiting time and increases the loss of jobs. The goal is to design better managing algorithms which take into account these two constraints to minimize : waiting time, loss rate and energy consumption. In [4] Mitrani studies the problem of managing a data center to keep a low energy consumption. This problem is modeled by a queue in which jobs can leave the system if the waiting time is too long. The proposed strategy is to consider some servers as a reserve group that are gradually switched on when the number of jobs in the buffer exceeds a certain threshold. Similarly these server groups are gradually switched off when the number of jobs in the buffer decreases and exceeds another threshold. The thresholds are analytically determined based on an objective function that takes into account the parameters of the systems.

In [2] we present, with other co-authors, a tool to study the trade-off between energy consumption and performance evaluation. The tool uses real traffic traces and stochastic monotonicity property to insure fast computation. Given a set of parameters that are fixed by the modeler, the tool determines the best threshold based policy. Our approach differs from the methods presented in [1, 4, 5] by several points. First it is numerical rather than analytical or simulation based. Thus, this paper considers less regular processes than the Poisson process considered by Mitrani. Note that the Markovian assumptions (Poisson arrivals, exponential services and switching times) and the infinite buffer capacity are not mandatory for this analysis. However, here, the arrival process is assumed to be stationary for short periods of time and change between periods. This allows us to represent for instance hourly or daily variations of the job arrivals. Real traffic traces are used to construct build discrete distributions for the job arrival.

2. Queuing model

In this work a data center is modeled by a discrete time queue with a finite buffer capacity and with a time slot equals to the sampling period used to sample the traffic traces. The job arrivals are specified by a discrete distribution. The system is analyzed for a finite time period (a day, a week). This time period is divided into sub-intervals where the batch of arrivals are supposed i.i.d. We design an optimization algorithm in order to manage energy consumption and QoS in the data center. The cost of the consumed energy depends on the number of operational servers. The QoS cost depends on the number of waiting time (which depends on the number of jobs in the buffer) and the losing rate (which depends on the number of lost jobs). Every slot, our algorithm minimizes an objective function that combines the cost of energy and the cost of QoS, in order to increase or decrease the number of operational servers according to traffic variation. As the model is solved numerically, it is much faster and more accurate than simulation. The work considers several tests for various types of arrivals : (i) arrivals with constant rate, (ii) arrivals defined by an i.i.d. discrete distribution, (iii) arrivals specified by a variable discrete distribution over time, (iv) and arrivals modeled by discrete distributions obtained from real traffic traces.

3. Tests & results

The algorithm is tested by numerical analysis under various types of job arrivals. We use real traffic traces to model arrivals, using the open *clusterdata-2011-2* trace [6], and we focus on the part that

contains the job events corresponding to the requests destined to a specific Google data center for the whole month of May 2011. The optimization algorithm that we suggest, adapts and adjusts dynamically the number of operational servers according to : traffic variation, workload, cost of keeping a job in the buffer, cost of losing a job, and energetic cost for serving a job. The job events are organized as a table of eight attributes ; we only use the column *timestamps* that refer to the arrival times of jobs expressed in micro-sec. This traffic trace is sampled with a sampling period equal to the slot duration. We consider frames of one minute to sample the trace and construct seven empirical distributions corresponding to arrivals during each day of the week. Such an assumption is consistent with the week evolution of job arrivals observed by long traces. These distributions have different statistical properties reflecting the fluctuation of traffic over the week. High arrivals rate is observed on Thursday, low arrivals rate on Saturday, Sunday and Monday, and medium arrivals rate during the rest of the week. For instance, we observe an average of 39 (resp. 58) jobs per minute during Sunday (resp. Thursday) with a standard deviation of 22 (resp. 38). Figure 1 shows the results of analyzing numerically the system whose parameters are :

Parameters	Value	Unit	Description
Max	100	servers	total number of servers
S	1	jobs/server	processing capacity of a server
B	300	jobs	buffer size

4. Conclusion

In this work we present an optimization stochastic algorithm in order to manage energy consumption and QoS in a data center modeled by discrete time queue. Every slot, the algorithm minimizes an objective function that combines the cost of energy and the cost of QoS, in order to change the number of operational servers according to traffic variation. We show the ability of our algorithm to adapt dynamically to arrivals changes. Test were shown through various numerical analysis for several types of arrivals, and specially for arrivals modeled by discrete distributions obtained from Google real traffic traces. The system starts turning on servers progressively when high arrivals rate is detected. And turn off gradually the servers when arrivals rate becomes low. Doing a closer analysis of the relationship between cost of energy, cost of QoS, workload and optimal number of operational servers is considered for future work to determine more accurate link between these parameters. We also intend to extend this study for the case in which, the number of served jobs in a slot by a server, is also defined by a distribution.

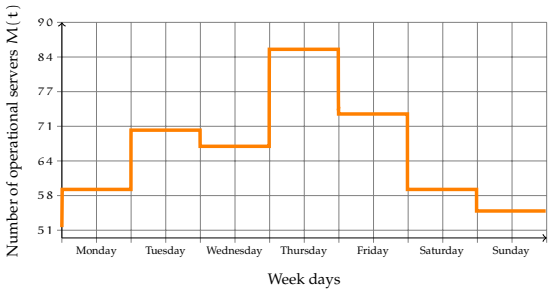

FIGURE 1 – Evolution, over a week, of the number of operational servers. Our algorithm adapts the number of operational server according to the traffic variation.

Bibliographie

1. K. Aidarov, Paul D. Ezhilchelvan, and Isi Mitrani. Energy-aware management of customer streams. *Electr. Notes Theor. Comput. Sci.*, 296 :199–210, 2013.
2. M. Bayati, M. Dahmoune, J.M. Fourneau, N. Pekergin, and D. Vekris. A tool based on traffic traces and stochastic monotonicity to analyze data centers and their energy consumption. In *Valuetools '15 : 9th international conference on Performance evaluation methodologies and tools*, page to appear. Acm, 2015.
3. Jonathan Koomey. Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, completed at the request of The New York Times*, page 9, 2011.
4. Isi Mitrani. Managing performance and power consumption in a server farm. *Annals OR*, 202(1) :121–134, 2013.
5. C. Schwartz, R. Pries, and P. Tran-Gia. A queuing analysis of an energy-saving mechanism in data centers. In *Information Networking (ICOIN), 2012 International Conference on*, pages 70–75, Feb 2012.
6. John Wilkes. More Google cluster data. Google research blog, November 2011. Posted at `http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html`.

Compter avec les Protocoles de Population

Yves Mocquard

Université de Rennes 1
Irisa
yves.mocquard@irisa.fr

1. Introduction

Le modèle des protocoles de population, introduit par Angluin et al. [1], fournit les bases théoriques pour analyser les propriétés émergeant d'un système constitué d'agents anonymes interagissant deux à deux. Nous définissons dans [2], un protocole de population permettant de compter exactement le nombre d'agents d'un type particulier. Nous montrons que ce protocole converge de manière logarithmique en temps distribué.
Beaucoup de travaux existent sur le problème de la majorité, Angluin et al. [4] ont mis au point un protocole rapide et simple, mais qui n'est exact que quand la majorité est large, ça reste un bon outil pour trouver un consensus. Alistarh et al. [4] ont récemment mis au point un protocole de majorité exact en en temps logarithmique mais avec une constante très grande.
Nous allons décrire un protocole de population qui résout le problème du comptage, lequel généralise celui de la majorité en l'affinant. En interrogeant n'importe quel agent, on peut connaître le nombre d'agent d'un type particulier en $O(\log n)$ interactions, avec une probabilité aussi grande que l'on souhaite.

2. Protocoles de Population

La définition qui suit est tirée de Angluin et al. [5].
Un protocole de population est caractérisé par un 6-uplet $(Q, \Sigma, Y, \iota, \omega, f)$, Σ est l'ensemble fini des symboles d'entrée, Y est l'ensemble fini des symboles de sortie, $\iota : \Sigma \to Q$ est la fonction d'entrée qui détermine l'état initial d'un agent, $\omega : Q \to Y$ est la fonction de sortie qui détermine le symbole de sortie d'un agent, et $f : Q \times Q \to Q \times Q$ est la fonction de transition qui décrit comment deux agents interagissent et mettent à jour leur état.
Au début, tous les agents démarrent avec un symbole initial venant de Σ. La fonction ι initialise l'état de chaque agent à partir de son symbole, puis au gré des interactions, les agents mettent à jour leur état utilisant la fonction de transition f.
Soit $C = \{C_t,\ t \geq 0\}$ un processus stochastique à temps discret avec comme ensemble d'états Q^n. Nous l'appellerons configuration un état de ce processus stochastique pour ne pas le confondre avec l'état des agents du protocole de population. Pour tout $t \geq 0$, la configuration à l'instant t de ce processus stochastique est notée par $C_t = (C_t^{(1)}, \ldots, C_t^{(n)})$, $C_t^{(i)}$ représente l'état de l'agent i à l'instant t.
A chaque instant t, deux indices distincts i et j sont successivement choisis parmi $1, \ldots, n$ avec la probabilité $p_{i,j}(t)$. Nous nommons X_t la variable aléatoire représentant ce choix, c'est-à-dire

$$\mathbb{P}\{X_t = (i, j)\} = p_{i,j}(t).$$

Nous considérons que les variables aléatoires X_t et C_t sont indépendantes.

3. Compter avec les Protocoles de Population

3.1. Introduction

Chaque agent possédant un symbole d'entrée issu de l'ensemble $\Sigma = \{A, B\}$, soit N_A et N_B le nombre d'agents ayant pour symbole respectivement A ou B, le but est de connaître la différence invariante $\kappa = N_A - N_B$. Et c'est ce que doit rendre la fonction de sortie ω.
Au début la fonction ι attribut à chaque agent l'état m ou $-m$ selon que son symbole soit respectivement, A ou B.
La fonction de transition consiste à attribuer à chaque agent la moyenne des deux valeurs.
Cet algorithme garde constant la somme de tous les états de chaque agent, somme qui est égale à $m(N_A - N_B)$.
La fonction de transition égalise progressivement l'état de tous les agents qui finissent par approcher de la même valeur $\frac{m(N_A - N_B)}{n}$. A partir de laquelle la fonction de sortie peut calculer $\kappa = N_A - N_B$.

3.2. Compter avec un ensemble d'états fini

Nous travaillons avec un ensemble fini d'états Q qui est un ensemble d'entier. Les paramètres Q, Σ, Y, ι et ω dépendent de l'application et seront définis à la fin de cette section pour le calcul de la différence invariante κ. La fonction de transition f est définie par

$$f(a, b) = \begin{cases} \left(\frac{a+b}{2}, \frac{a+b}{2}\right) & \text{si } a + b \text{ est pair} \\ \left(\frac{a+b-1}{2}, \frac{a+b+1}{2}\right) & \text{si } a + b \text{ est impair} \end{cases}$$

Une fois que le couple (i,j) est choisi à l'instant t, C_{t+1} est défini par

$$\left(C_{t+1}^{(i)}, C_{t+1}^{(j)}\right) = \begin{cases} \left(\frac{C_t^{(i)}+C_t^{(j)}}{2}, \frac{C_t^{(i)}+C_t^{(j)}}{2}\right) \text{ si } C_t^{(i)} + C_t^{(j)} \text{ pair} \\ \left(\frac{C_t^{(i)}+C_t^{(j)}-1}{2}, \frac{C_t^{(i)}+C_t^{(j)}+1}{2}\right) \text{ si } C_t^{(i)} + C_t^{(j)} \text{ impair} \end{cases}$$

$$\text{et } C_{t+1}^{(m)} = C_t^{(m)} \text{ pour } m \neq i,j.$$

Le lemme suivant est fondamental, il établit que la somme des états des agents reste invariante.

Lemme 1 *Pour tout $t \geq 0$, nous avons*

$$\sum_{i=1}^{n} C_t^{(i)} = \sum_{i=1}^{n} C_0^{(i)}.$$

Nous notons ℓ la moyenne des coordonnées de C_t et L le vecteur de $\mathbb{R}^n$ dont toutes les coordonnées sont égales à ℓ, c'est-à-dire

$$\ell = \frac{1}{n}\sum_{i=1}^{n} C_t^{(i)} \text{ et } L = (\ell, \ldots, \ell).$$

Théorème 2 *En supposant un choix uniforme du couple (i,j), c'est-à-dire si, pour $i \neq j$,*

$$p_{i,j}(t) = \frac{1}{n(n-1)},$$

alors nous avons

$$\mathbb{E}\left(\|C_t - L\|^2\right) \leq \left(1 - \frac{1}{n-1}\right)^t \mathbb{E}\left(\|C_0 - L\|^2\right) + \frac{n}{4}.$$

En utilisant l'inégalité de Markov on obtient le corollaire suivant qui donne une δ-approximation de l'écart maximum entre les coordonnées de C_t et ℓ.

Corollaire 3 *Pour tout $\delta \in]0,1[$, s'il existe une constante K telle que $\mathbb{E}(\|C_0 - L\|_\infty) \leq K$ alors, pour tout $t \geq (n-1)\ln(4K^2)$ nous avons*

$$\mathbb{P}\{\|C_t - L\|_\infty \geq \sqrt{\frac{n}{2\delta}}\} \leq \delta.$$

Nous pouvons maintenant appliquer ces résultats au calcul de la différence invariante κ. L'ensemble des entrées est $\Sigma = \{A, B\}$, et la fonction d'entrée ι est définie par $\iota(A) = m$ et $\iota(B) = -m$, m étant un entier strictement positif. Cela signifie que, pour tout $i = 1, \ldots, n$, $C_0^{(i)} \in \{-m, m\}$. Nous avons

$$\ell = \frac{1}{n}\sum_{i=1}^{n} C_0^{(i)} = \frac{\kappa m}{n},$$

ce qui montre à partir du Lemme 1 que κ est invariant. L'ensemble des états Q est maintenant l'ensemble $\{-m, -m+1, \ldots, m-1, m\}$. La fonction de sortie est, pour tout $x \in Q$,

$$\omega(x) = \lfloor nx/m + 1/2 \rfloor.$$

L'ensemble de sortie Y est l'ensemble des valeurs possibles de κ, i.e. $Y = \{-n, -n+1, \ldots, n-1, n\}$.

Théorème 4 *Pour tout $\delta \in]0,1[$, en prenant $m = \left\lceil \sqrt{2}n^{3/2}/\sqrt{\delta} \right\rceil$ et pour tout $t \geq (n-1)\left(5\ln 2 + 3\ln n - \ln\delta + \frac{2}{m-1}\right)$, nous avons*

$$\mathbb{P}\{\omega(C_t^{(i)}) = \kappa, \text{ pour tout } i = 1, \ldots, n\} \geq 1 - \delta.$$

Donc le temps de convergence pour obtenir κ avec une probabilité aussi grande que l'on veut est $O(n \log n)$ et ainsi le temps de convergence parallèle pour obtenir κ avec une probabilité aussi grande que l'on veut est $O(\log n)$.
Ce protocole nécessite la connaissance préalable du nombre d'agents n.
Dans le futur nous comptons proposer un protocole qui n'a plus cette nécessité. Deux voies sont possibles, la première consiste à calculer la différence en terme de pourcentage, la deuxième consiste, en utilisant une élection de leader, à calculer, au préalable, la taille du système.

Bibliographie

1. Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fisher, and René Peralta. Conputation in netWorks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235-253,2006.
2. Yves Mocquard, Emmanuelle Anceaume, James Aspnes, Yann Busnel, and Bruno Sericola. Counting with Population Protocols. In *Proceedings of the 14th IEEE International Symposium on Network Computing and Applications (IEEE NCA15)*, September 2015.
3. Dan Alistarh, Rati Gelashvili, and Milan Vojvonic. Fast and exact majority in population protocols. Technical report, Microsoft Research, 2015.
4. Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocols for fast rubust approximate majority. 20(4) :279-304, 2008.
5. Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computationnal power of population protocols. *Distributed Computing*, 20(4) :278-304,2007.

Evaluation des performances bout en bout du trafic TCP sous le régime "Équité Équilibrée"

Jean Marie Garcia [1], Mohamed El Hedi Boussada [2]

[1] LAAS-CNRS, SARA
7 avenue du Colonel Roche, 31077
Toulouse,France
jmg@laas.fr

[2] SUP'COM, MEDIATRON
Technopôle El Gazala, 2088 Ariana, Tunisie
med.elhadi.boussada@supcom.tn

1. Introduction

Aujourd'hui, la plupart du trafic circulé dans l'Internet est généré par un transfert de documents comme les fichiers ou les pages Web [1]. Ce trafic est élastique dans le sens où la durée de chaque transfert dépend de l'etat du réseau.
Chaque document est divisé en une séquence de paquets, appelé flux, dont le débit d'envoi est adapté selon l'état de congestion dans le réseau, généralement sous le contrôle du protocole TCP. La qualité du transfert dépend alors du temps nécessaire pour transférer avec succés tous les paquets du flux. En ce sens, les performances du trafic élastique se manifestent principalement au niveau flux et peuvent être traduits par le débit moyen de chaque flux de données [3].
Comme il y a plusieurs classes des flux, l'évolution du nombre des flux pour chaque classe dépend toujours de la nature de l'allocation des ressources. La plupart des travaux sont concentrés sur des allocations basés sur des fonctions d'utilités comme l'allocation classique ≪ équité max-min ≫ et ≪ l'équité proportionnelle de Kelly ≫ . En général, l'analyse d'un réseau fonctionnant sous régime de ces allocations est assez difficile. Une des raisons est qu'ils ne conduisent pas à une expression explicite pour la distribution stationnaire, qui détermine le nombre typique de flux concurrents de chaque classe[2]. Dans ce contexte, La notion de l'équité équilibré (Balanced Fairness en anglais) a été introduite par Bonald et Proutiére comme un moyen d'évaluer approximativement la performance de ces allocations équitables[1]. Une propriété clé de l'équité équilibrée est son insensibilité : la distribution stationnaire de nombre des flux est indépendante de toutes les caractéristiques fines du trafic[2]. La seule hypothése requise est que les flux arrivent comme un processus de Poisson, qui est effectivement satisfaite dans la pratique des grands réseaux avec de nombreux abonnés.
Toutefois, l'équité équilibrée reste complexe à utiliser dans un contexte pratique car elle requiert le calcul de la probabilité de chacun des états possibles du système, et est donc confrontée à l'explosion combinatoire de l'espace d'états pour de grands réseaux. Dans ce contexte, il est primordial de proposer des solutions (ou bien des approximations) permettant de calculer efficacement les métriques de performance sans nécessiter l'évaluation des probabilités individuelles des états.
Ce papier vise à proposer des approximations simples et explicites pour évaluer les performances de bout en bout des flux élastiques sous le régime d'équité équilibrée.

2. Modèle

Le modèle consiste en un ensemble de L liens où chaque lien l a une capacité C_l.Un certain nombre de flots élastiques sont en compétition pour le partage de la bande passante de ces liens. Soit E l'ensemble de classes de ces flots. Chaque flux d'une classe $i \in E$ est caractérisé par leur débit maximum noté d_i et la taille moyenne de fichier à transférer noté σ_i.La congestion force les flux à réduire leurs débits et par suite d'augmenter le délai du transfert.
Les flux arrivent suivant un processus de Poisson de moyenne λ_i pour les flux de classe $i \in E$. L'intensité du trafic d'une classe i est donné par le produit $\rho_i = \lambda_i \sigma_i$. On désigne par A la matrice d'incidence définit comme :$a_{il} = 1$ si les flots de classe i utilisent le lien $l \in L$, et 0 sinon.
Soit $\theta_l = \sum_{i \in E} a_{il} \rho_i$ le trafic offert à un lien $l \in L$. Pour maintenir la stabilité du système, on suppose que la charge totale de chaque lien $l \in L$ est strictement inférieure à sa capacité : $\theta_l < C_l$.
On note par x_i le nombre des flux présents dans le réseau pour la classe i et on désigne par $x = (x_i)_{i \in E}$ l'état du réseau.
Pour la suite, on évalue les performances du trafic élastique à travers le débit moyen de chaque flux. En utilisant la formule de Little, le débit moyen de chaque flux d'une classe $i \in E$ est donné par :

$$\gamma_i = \frac{\rho_i}{E[x_i]} \tag{1}$$

où $E[x_i]$ est le nombre moyen des flux de la classe i.

3. Analyses

On partage les ressources du réseau selon le régime équité équilibrée.

3.1. Cas d'un seul lien

On suppose ici que le réseau est limité à un lien unique de capacité C. On note $\theta = \sum_{i \in E} \rho_i$ le trafic offert à ce lien.
La probabilité de congestion d'une classe $i \in E$ est donnée par la probabilité de l'ensemble $B_i = \{x : C - d_i < n\}$ où $n = \sum_{k \in E} d_k x_k$:

$$\pi(B_i) = \frac{1}{C-\theta} \sum_{k \in E} \rho_k \pi(W_k) + \pi(W_i) \qquad (2)$$

Où $W_k = \{x : C - d_k < n \leq C\} \quad \forall k \in E$.

$$\pi(W_k) = \pi(0) \sum_{C-d_k < n \leq C} \prod_{j \in E} \frac{(\frac{\rho_j}{d_j})^{x_j}}{x_j!} \qquad (3)$$

Avec :

$$\pi(0) = \Bigg(\sum_{C-d_k < n \leq C} \prod_{j \in E} \frac{(\frac{\rho_j}{d_j})^{x_j}}{x_j!} + \frac{1}{C-\theta} \sum_{k \in E} \rho_k \sum_{C-d_k < n \leq C} \prod_{j \in E} \frac{(\frac{\rho_j}{d_j})^{x_j}}{x_j!} \Bigg)^{-1} \qquad (4)$$

Le nombre moyen des flux d'une classe $i \in E$ est donné par :

$$E[x_i] = \frac{\rho_i}{d_i} \Bigg[1 - \pi(B_i) + \frac{1}{C-\theta} \pi(0) \sum_{k \in E} \rho_k \sum_{C-d_i-d_k < n \leq C-d_i} \prod_{j \in E} \frac{(\frac{\rho_j}{d_j})^{x_j}}{x_j!} \Bigg] + \frac{\rho_i}{C-\theta} \pi(B_i) \qquad (5)$$

On montre que :

$$E[x_i] \leq \frac{\rho_i}{d_i} + \frac{\rho_i}{C-\theta} \pi(B_i) \qquad (6)$$

L'inéquation (6) donne une borne supérieure au nombre moyen des flux de chaque classe. Bien qu'il soit difficile de le démontrer mathématiquement, toutes nos observations numériques concordent sur le fait que cette borne supérieure fournit une très bonne approximation de $E[x_i]$.Donc on peut écrire :

$$E[x_i] \approx \frac{\rho_i}{d_i} + \frac{\rho_i}{C-\theta} pi(B_i) \qquad (7)$$

3.2. Généralisation vers le cas d'un réseau

Pour le cas d'un réseau, on propose l'approximation suivante :

$$E[x_i] \approx \frac{\rho_i}{d_i} + \sum_{l \in L} a_{il} \pi(B_i^l) \frac{\rho_i}{C_l - \theta_l} \qquad (8)$$

Où :

$$\pi(B_i^l) = \frac{1}{C_l - \theta_l} \sum_{k \in E} a_{kl} \rho_k \pi(W_k^l) + \pi(W_i^l) \qquad (9)$$

Pour tout $i \in E$,les ensembles B_i^l et W_i^l (ainsi leurs probabilités) sont définis de la méme façon que la section précedente en remplacant C par C_l, θ par θ_l et n par $n_l = \sum_{k \in E} a_{kl} d_k x_k$.
La validité de cette approximation a été prouvée par simulations sur NS-2 où l'erreur relative n'a pas dépassé le 5% pour des petites et moyennes valeurs de θ_l.

4. Conclusion

Malgré son efficacité dans l'étude des performances du trafic élastique, l'allocation équité équilibrée reste complexe à utiliser dans un contexte pratique, et surtout pour des réseaux assez larges. Dans ce sens, on a proposé des approximations pour évaluer le débit moyen de bout-en-bout du trafic élastique sous une telle allocation. Les approximations données sont précises et permettent une généralisation pour de grands réseaux avec un temps de calcul raisonnable. Ces résultats conduisent directement à des règles simples d'ingénierie de trafic et à des méthodes robustes d'évaluation des performances nécessaires à la maîtrise des réseaux actuels.

Bibliographie

1. Bonald, T., Virtamo, J. (2004). Calculating the flow level performance of balanced fairness in tree networks. Performance Evaluation, 58(1), 1-14.
2. Bonald, T., Haddad, J. P., Mazumdar, R. R. (2011, September). Congestion in large balanced multirate links. In Proceedings of the 23rd International Teletraffic Congress (pp. 182-189). International Teletraffic Congress.
3. Brun, O., Al Sheikh, A., Garcia, J. M. (2009, September). Flow-level modelling of TCP traffic using GPS queueing networks. In Teletraffic Congress, 2009. ITC 21 2009. 21st International (pp. 1-8). IEEE.

Average complexity of the Best Response Algorithm in Potential Games

Stéphane Durand and Bruno Gaujal

Univ. Grenoble Alpes
Inria
stephane.durand@inria.fr
bruno.gaujal@inria.fr

1. Introduction

The computation of Nash Equilibria (NE) in games has been investigated of many papers. The most general result is in [2] and says that the problem of computing NE is PPAD complete.

Potential games have been introduced in [7] and have proven very useful, especially in the context of routing games, first mentioned in [1] and exhaustively studied ever since, in the transportation as well as computer science litterature, see for example [6]. For potential games, efficient polynomial time algorithms exist in symmetric cases (see [4]). However, the same paper shows that the computation of NE for general potential games is PLS complete. The Best Response Algorithm (BRA) is probably the most popular algorithm that converges to a pure Nash equilibrium (NE) in potential games [5]. However, its complexity has attracted surprisingly little attention.

In this paper, we analyze the performance of BRA over a potential game with N players, each with A possible actions. We show that on average, the Best Response Algorithm takes $\log(N)+e^{\gamma}$ (γ is the Euler constant) effectives steps and makes $e^{\gamma}AN$ comparisons before finding a NE.

These numbers say that BRA is very efficient on average to compute NE, even if this is a PLE complete problem. Our analysis is based on two ingredients, one is the construction of an approximation of the behavior of BRA, where each state is examined at most once and the second is the use of a continuous space discrete time Markov chain to analyze the average complexity.

2. Best Response Algorithm and Potential games

We consider a game with a finite number N of players and a finite strategy space for each player, each of size A, and the corresponding utility functions. The game $\mathfrak{G} \stackrel{\text{def}}{=} \mathfrak{G}(\mathcal{N}, \mathcal{A}, u)$ will be a tuple consisting of

- a finite set of *players* $\mathcal{N} = \{1, \dots, N\}$;
- a finite set $\mathcal{A}_k$ of *actions* (or *pure strategies*) for each player $k \in \mathcal{N}$; The set of *action profiles* or *states* of the game is $\mathcal{A} \stackrel{\text{def}}{=} \prod_k \mathcal{A}_k$;
- the players' *payoff functions* $u_k : \mathcal{A} \to \mathbb{R}$.

We define the classical *best response correspondence* $\mathbf{br}_k(x)$ as the set of all actions that maximizes the payoff for player k under profile x :

$$\mathbf{br}_k(x) \stackrel{\text{def}}{=} \left\{ \underset{\alpha \in \mathcal{A}_k}{\operatorname{argmax}}\, u_k(\alpha; x_{-k}) \right\}. \qquad (1)$$

A *Nash equilibrium* (NE) is a fixed point of the correspondence, i.e. a profile x^* such that $x^*_k \in \mathbf{br}_k(x^*)$ for every player k.

Definition 1 (Potential games) *A game is a* potential game *[5] if it admits a function (called the potential)* $\Phi : \mathcal{A} \to \mathbb{R}$ *such that for any player* k *and any* unilateral *deviation of* k *from action* α *to* α', $u_k(\alpha, x_{-k}) - u_k(\alpha', x_{-k}) = \Phi(\alpha, x_{-k}) - \Phi(\alpha', x_{-k})$.

We consider a version of *Best Response Algorithm* (BRA) where the next player is selected according to a round robin pattern. Other patterns can also be considered using the same approach and can be shown to have a similar behavior.

Algorithm 1: Best Response Algorithm (BRA)

Input :
Game utilities $(u_i(\cdot))$,
Initial state $(x(0))$,
Infinite seq. of players $R = (1, 2, \dots, N, 1, \dots)$.

foreach *player* $k \in K$ **do**
  $\mathrm{stop}_k := \mathtt{false}$
repeat
  Pick next player $k := R_{t+1}$
  Select new action $\alpha_k := \mathbf{br}_k(x(t))$
  $\mathrm{stop}_k := \mathbf{1}_{\{\alpha_k = x_k(t)\}}$;
  $x_k(t+1) := \alpha_k$;
until $\mathrm{stop}_1 \wedge \mathrm{stop}_2 \wedge \cdots \wedge \mathrm{stop}_N$;

A famous result first proved in [5] states that for any potential game $\mathfrak{G}$, Algorithm 1 converges in finite time to a Nash Equilibrium of $\mathfrak{G}$.

3. Complexity

Let us consider three complexity measures (related to each other) : T_{BRA} is the number of iterations

(or the number of times that the function **br** was called) before BRA reaches a Nash equilibrium. The total number of comparisons is denoted C_{BRA}. One should expect that $C_{BRA} \approx (A-1)T_{BRA}$. Finally, the number of different states visited by BRA is denoted M_{BRA}. Of course, $M_{BRA} \leqslant T_{BRA}$. The proofs of Theorems 1 and 2 are not provided due to lack of space. They are available in a research report [3].

Theorem 1 *In the worst case, under round robin revisions,* $T_{BRA} = NA^{N-1}$.

3.1. Randomization

In the following we will analyze the average complexity of BRA.

We randomize over all the potential games over which BRA is used. Since the behavior of BRA only depends on the potential function, we randomize directly over the potential Φ. The natural randomization is to consider all possible total orderings of the set $\{\Phi(x), x \in \mathcal{A}\}$ (there are $(A^N)!$ of them) and pick one uniformly. This is equivalent to pick iid potentials in all states, uniformly distributed in $[0,1]$.

3.2. Markovian Analysis

We will be analyzing the intersection-free approximation of the behavior of BRA (where no state is visited twice) whose behavior is asymptotically the same as BRA.

Let y be the potential of the current state x : ($y \stackrel{\text{def}}{=} \Phi(x)$). If $k-1$ players have already played best response without changing the state, then the evolution at the next step of BRA is as follows. The k-th player computes its best response. This player has $a \stackrel{\text{def}}{=} A-1$ new actions whose potential must be compared with the current potential (y). With probability y^a none of the new actions beat the current choice. The state remains at y and it is the turn of the $k+1$-st player to try its best response. With probability $1-y^a$, one of the new actions is the best response. The current state moves to a new state with a larger potential and the number of players for which the new state is a best response is set back to 1.

This says that the couple (Y_t, K_t) is a Markov chain, where Y_t is the potential at step t, in $[0,1]$ and K_t is the current number of players whose best response did not change the current state (in $\{1,2,\dots,N\}$). Its transitions are :

$$\mathbb{P}\Big((Y,K)_{t+1} = (y,k+1)\Big|(Y,K)_t = (y,k)\Big) = y^a,$$

and, if $z > y$,

$$\mathbb{P}\Big((Y,K)_{t+1} \in ([z,1],1)\Big|(Y,K)_t = (y,k)\Big) = 1-z^a.$$

Let $C(y,k)$ be the average number of comparisons required to reach a NE, starting in a state with potential y where k players have played without changing their action. The forward equation for $C(y,k)$ is :

$$\begin{aligned} C(y,k) =& y^a(C(y,k+1)+a) \\ &+ \int_y^1 a u^{a-1}(C(u,1)+a)du, \end{aligned}$$

with the boundary conditions $C(1,1) = a(N-1)$ and $C(y,N) = 0$.

Solving these equations leads to the following proposition (quantities M_{BRA} and T_{BRA} are analyzed similarly).

Theorem 2 *The average number of moves in BRA verifies* $\mathbb{E}M_{BRA} \leqslant \log(N) + e^\gamma + O(1/N)$.
The average number of comparisons verifies
$\mathbb{E}C_{BRA} \leqslant e^\gamma(A-1)N + o(A)$
and the average number of steps verifies
$\mathbb{E}T_{BRA} \leqslant e^\gamma N + o(1)$.

Bibliographie

1. M. Beckman, C. B. McGuire, and C. B.Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
2. P.W. Goldberg C. Daskalakis and C.H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(3) :195–259, 2009.
3. Stéphane Durand and Bruno Gaujal. Efficiency of the best response algorithm in potential games. Technical report, Inria, 2015.
4. Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612. ACM, 2004.
5. Dov Monderer and Lloyd Shapley. Potential games. *Games and economic behavior, Elsevier*, 14(1) :124–143, 1996.
6. A. Orda, R. Rom, and N. Shimkin. Competitive routing in multeuser communication networks. *IEEE/ACM Trans. on Networking*, 1(5) :510–521, 1993.
7. Robert W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *Int. J. of Game Theory, Springer*, 2(1) :65–67, 1973.

Optimal adaptive routing in packet-switched networks

Bruno Gaujal and Baptiste Jonglez

Univ. Grenoble Alpes
Inria
ENS Lyon
baptiste.jonglez@ens-lyon.org
bruno.gaujal@inria.fr

1. Introduction

Communication networks are becoming increasingly multipath and a common challenge is to exploit this path diversity. More precisely, the problem can be modelled as a *multi-commodity flow problem* : Given a number of concurrent source-destination flows, the problem is to assign these flows to network paths, while respecting capacity constraints.

Here, we present a novel algorithm for adaptive routing in arbitrary network topologies, mapping source-destination flows to paths. We claim that it provides a viable and stable solution to adapt to traffic conditions, and effectively avoids congestion. Our algorithm is based on theoretical grounds from game theory, while our implementation leverages SDN protocols to ease deployment.

On the theoretical side, our distributed routing algorithm is endowed with the following desirable properties for efficient implementation :

- It is fully distributed without any information sharing ;
- It is oblivious to the network topology ;
- It only uses on-line and local information ;
- There are no endless oscillations and it is numerically stable ;
- It is robust to out-dated information and measurement errors ;
- It does not require time synchronization between routers ;
- and it converges fast even if the number of flows is very large.

2. Routing Algorithm

Let us consider a routing problem in a communication network. Several *flows* of packets must be routed over a communication network. The topology of the network is fixed but arbitrary.

Each flow $k \in \mathcal{K}$ is characterized by a source-node, a destination-node and a nominal arrival rate of packets, λ_k. Also, each flow is affected a set $\mathcal{P}_k$ of paths in the network from its source to its destination, made of P_k paths. A *configuration* is a choice of one path per flow. The delay over each link and each node in the network depends on the the load on the link (node), in an unspecified manner. For one flow, say k, we denote by $d_k(p_1, \cdots, p_k, \cdots, p_K)$ the end to end *average delay* experienced by packets of flow k under the configutration where flow 1 uses path p_1, flow 2 uses path p_2, and so forth.

The following algorithm is run by each flow k, independently. It is probabilistic and maintains two vectors of size P_k. The *probabilistic choice* vector, $\mathbf{q}_k = (q_1 \dots q_{P_k})$ gives at each step the probabilities to choose the paths and the *score vector* $\mathbf{Y}_k = (Y_1 \dots Y_{P_k})$ that attributes a score to the paths.

The main loop of the algorithm is as follows (index k is skipped).

At each local clock tick, a path p is chosen according to $\mathbf{q}$, and packets are sent along p. The average delay of packets over this path is measured. The score Y_p is updated according to a discrete dynamics inspired from game theory and in turn, the probability vector is modifed for the next path selection. This repeats forever, or until a stable path has been reached for all flows, *i.e.* $\mathbf{q}$ becomes a degenerate probability vector (all coordinates are zero but one) for all flows. The algorithm uses 3 parameters : τ is a discounting factor over past scores, $(\gamma_n)_{n\in\mathbb{N}}$ is a vanishing sequence of step sizes and $(\beta_n)_{n\in\mathbb{N}}$ is a sequence of bounding terms controlling the growth rate of the scores. In the algorithm, $\wedge$ denotes the minimum operator.

Algorithm 1: OPS : Online Path Selection for k

Initialize :
$n \leftarrow 0$; $\mathbf{q} \leftarrow (\frac{1}{P}, \dots, \frac{1}{P})$; $\mathbf{Y} \leftarrow (0, 0, \dots, 0)$;
repeat
  When local clock ticks for the nth time ;
  $n \leftarrow n + 1$;
  select new path p w.r.t. probability vector $\mathbf{q}$;
  Use path p and measure its delay D ;
  Update score of p :
  $$Y_p \leftarrow \left(Y_p - \gamma_n (D + \tau Y_p)/q_p \right) \wedge \beta_n \, ;$$
  update proba. : $\forall s \in \mathcal{P}_k, q_s \leftarrow \frac{\exp(Y_s)}{\sum_\ell \exp(Y_\ell)}$;
until *end of time*;

Theorem 1 (Convergence to equilibirum)
Under mild technical assumptions, for all $\epsilon > 0$, there exist $\tau > 0$ such that the algorithm converges to an ϵ-optimal configuration, in the following sense :

For each flow ℓ, the probability vector $\mathbf{q}$ *converges to an almost degenerate probability :* q_p *becomes smaller than* ϵ *for all* $p \in \mathcal{P}_\ell$ *except for one path, say* p^*_ℓ*, for which it grows larger than* $1-\epsilon$.
Furthermore, after convergence, no flow can reduce its delay : $\forall p' \in \mathcal{P}_\ell$,
$d_\ell(p^*_1, \ldots, p^*_\ell, \ldots, p^*_K) \leqslant d_\ell(p^*_1, \ldots, p', \ldots, p^*_K)$.

The proof is based on a general convergence theorem from game theory, proved in [1]. It is essentially based on two facts. The empirical delay D measured on packets using path p for flow f has no bias, conditionally on the past : At step n, $\mathbb{E}(D|\mathcal{F}_n) = d_k(p_1, \ldots, p_K)$. This implies that the scores $\mathbf{Y}$ form a stochastic approximation of a continuous deterministic dynamics that converges to Nash Equilibria in all potential games.

3. Implementation and experimental Results

Looking at our OPS algorihtm, we can note that it is completely distributed : it requires only local measures, local choices, and no coordination is needed between routers. This eases implementation. The only difficulty lies in the ability to select paths from source to destination : this is not practical in current next-hop forwarding networks. Our implementation is based on an equivalent version of OPS, where one gateway router makes a choice among all possible next-hop routers for each of its flow. Thus, the local actions of several routers between source and destination implicitely determine the path from source to destination.

Our implementation takes the form of an Openflow controller, using the Ryu library. The routing table of each gateway router is programmed and constantly updated by a dedicated controller. Furthermore, each gateway router sends packet headers to its controller, for delay computation.

To run realistic experiments, we use Mininet [2], a widely adopted network emulator. Mininet is used to build a virtual network topology, thanks to the *network namespaces* feature of the Linux kernel. On this virtual topology, we can run our implementation exactly as if we had a real network at hand.

To validate our approach, we use a simple topology, with two gateway routers and three hosts. Two hosts are simply connected to their gateway, while the third host can be reached via two different paths. The topology is shown in Figure 1.

We consider two TCP flows from host 1 and host 2, both destinated to host 3. Each flow is restricted by the sender to use no more than 8 Mbit/s, to avoid saturating all links whatever the choice of flow assignment. If both flows are forwarded over the same rightmost link, congestion will occur, because of the limited capacity.

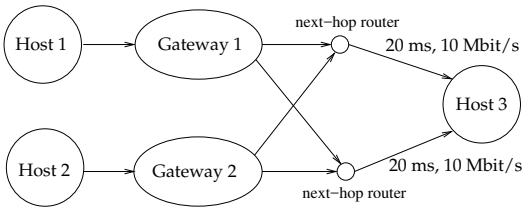

FIGURE 1 – Simple topology, where the two rightmost links have limited capacity (10 Mbit/s) and a latency of 20 ms. Both host 1 and host 2 send a flow to host 3.

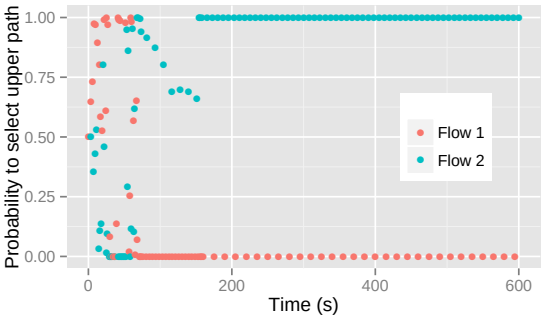

FIGURE 2 – Probability over time to select the upper path for each flow.

Figure 2 shows our experimental results. For each flow, the probability to use the upper path is plotted over time. After an initial period of exploration, the gateway routers converge to a stable state, in which each path to host 3 supports a single flow.

Bibliographie

1. Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. Penalty-Regulated Dynamics and Robust Learning Procedures in Games. *Mathematics of Operations Research*, 40(3) :611–633, 2015.
2. Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop : rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM, 2010.

A Mean-Field Game with Explicit Interactions for Epidemic Models

Josu Doncel, Nicolas Gast, Bruno Gaujal *

INRIA Grenoble Rhônes-Alpes
655 Avenue de l'Europe
38330 Montbonnot-Saint-Martin
{josu.doncel, nicolas.gast,bruno.gaujal}@inria.fr

1. Introduction

Game theory studies the rational behavior of decision-makers (called players in the following). A crucial notion is the concept of Nash equilibrium. A Nash equilibrium is an allocation of strategies such that no player can benefit from unilateral deviation. Although any finite game has at least one Nash equilibrium, it is shown in [1] that computing a Nash equilibrium is a PPAD-complete [2] problem. This suggests that the computation of a Nash equilibrium is not tractable when the number of players or of strategies is large. As an alternative, the notion of mean-field games has been introduced by Lasry and Lions in [6], which is a game where an individual object is infinitesimal and does not affect the global system behavior.

Here, we study mean field games with two specific features : Each player has a finite state space (instead of a continuous one in [6]), and the dynamics of one player depends explicitly on the behavior of the others (unlike in most works in mean field games [6, 3]). The general theory is developed in [2]. In this document, we illustrate the theory with a vaccination problem.

2. Model Description

2.1. Epidemic Model

We consider a population of N homogeneous objects that evolve in continuous time from 0 to T. The objects can be susceptible, infected, recovered or vaccinated. We denote by $S(t)$, $I(t)$, $R(t)$ and $V(t)$ the proportion of the population that is, respectively, susceptible, infected, recovered and vaccinated at time t.

The dynamics of one object is a Markov process that can be described as follows. An object encounters other objects with rate β. If the initial object was susceptible and the encounter was infected, the first object becomes infected. An infected object recovers at rate γ. We also consider that there is a vaccination policy $\mathbf{b}$ that is applied to each object of the susceptible population. The vaccination rate $\mathbf{b}$ is a function from 0 to T that takes values in the interval $[0, b_{max}]$. Once an object is vaccinated or recovered, it does not change its state. The dynamics of an object is described in Figure 1.

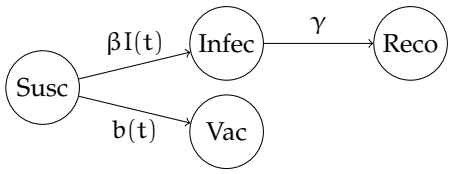

FIGURE 1 – The dynamics of an object in the epidemic model.

We are interested in the analysis of this epidemic model for a large number of objects. When $N \to \infty$, the dynamics of the population converges [4] to the following system of differential equations :

$$\begin{cases} \dot{S}(t) = -\beta \cdot S(t) \cdot I(t) - b(t) \cdot S(t) \\ \dot{I}(t) = \beta \cdot S(t) \cdot I(t) - \gamma \cdot I(t) \\ \dot{R}(t) = \gamma \cdot I(t) \\ \dot{V}(t) = b(t) \cdot S(t) \end{cases} \tag{1}$$

In [5] the authors develop an approximation of this epidemic model and characterize the solution of the derived mean-field game. In the rest of the paper, we show that the mean-field game corresponding to this model is tractable and can be analyzed rigorously.

2.2. Mean-Field Game

We focus on a particular object, that we call Player 0. Let $X(t) \in \{\mathrm{Sus}, \mathrm{Infec}, \mathrm{Reco}, \mathrm{Vac}\}$ be the state of Player 0 at time t. We note that the evolution of $X(t)$ depends on the infected population. We assume that the rest of the population applies a fixed vaccination policy $\mathbf{b}$. Player 0 chooses its vaccination policy $\mathbf{b}_0$, so as to minimize its expected individual cost, which is

$$C_{ind}(\mathbf{b}_0, \mathbf{b}) = \int_0^T (c_V b_0(t) \mathbb{P}(X(t) = \mathrm{Susc}) + c_I \mathbb{P}(X(t) = \mathrm{Infec}))dt,$$

∗. This work is partially supported by the EU project QUANTICOL, 600708.

2. PPAD stands for "polynomial parity arguments on directed graphs". It is a complexity class that is a subclass of NP and is believed to be strictly greater than P.

where c_V is the vaccination cost and c_I is the unit time cost of being infected.
We call the *best response to* $\mathbf{b}$ and denote by $BR(\mathbf{b})$ the set of vaccination policies that minimize the cost of Player 0 for a given vaccination policy of the population $\mathbf{b}$:

Definition 1 (Best-Reponse)

$$BR(\mathbf{b}) = \arg\min_{\mathbf{b_0}} C_{ind}(\mathbf{b}_0, \mathbf{b}). \quad (2)$$

We now define the notion of a mean-field equilibrium for this game. It is a vaccination strategy $\mathbf{b}^{MFE}$ such that when the population chooses the vaccination policy $\mathbf{b}^{MFE}$, a selfish Player 0 would also choose the same vaccination policy $\mathbf{b}^{MFE}$:

Definition 2 (Symmetric Mean-Field Equilibrium) *The vaccination policy $\mathbf{b}^{MFE}$ is a symmetric mean-field equilibrium if and only if*

$$\mathbf{b}^{MFE} \in BR(\mathbf{b}^{MFE}).$$

The rationale behind this definition is when one considers that the population is made of players that each take self-interested decisions. As the population is homogeneous, each object best-response is the same as Player 0. In other words, for a given population vaccination policy $\mathbf{b}$, all the objects of the populations choose the strategy $BR(\mathbf{b})$. A mean-field equilibrium is a situation where no object has incentive to deviate unilaterally from its strategy.

2.3. Centralized Control Problem

The mean-field game scenario corresponds to a case where the decisions are selfish and decentralized. The corresponding centralized control problem can also be defined naturally. For a given vaccination population $\mathbf{b}$,the average system cost is

$$C_{sys}(\mathbf{b}) = \int_0^T (c_V\ b(t)\ S(t) + c_I\ I(t))\ dt.$$

This cost represents the cost in the system when the population vaccination policy is $\mathbf{b}$. A global optimum is a vaccination policy that minimizes the system cost

Definition 3 (Global Optimum)

$$\mathbf{b}^{OPT} \in \arg\min_{\mathbf{b}} C_{sys}(\mathbf{b}). \quad (3)$$

3. Main Results

The mean-field game we analyze here is a particular case of the model of [2]. In [2], we introduce the mean-field games with explicit interactions, which is a discrete state space model where the transition rates between states depend not only on the actions taken, but also on the empirical measure of the system. This *explicit interactions* between objects makes our model distinct from most work on mean-field games.
To characterize a symmetric mean-field equilibrium of the epidemic model, we model the best-response of the generic object as a Continuous Time Markov Decision Process and we show that, for any population vaccination policy $\mathbf{b}$, the best-response strategy of the generic object is of threshold type. This result yields the following proposition.

Proposition 1 *There exists a symmetric mean-field equilibrium that is pure and of threshold type.*

We also analyze the global optimum of the epidemic model.

Proposition 2 *There exists a global optimum of threshold type.*

Unfortunately, in all but degenerated cases, the thresholds do not coincide, so that the price of anarchy of this model is never equal to 1. Numerical simulations show that the price of anarchy is small in general. A pricing mechanism can be used to force the equilibrium to coincide with the global optimum. Our numerical experiments show that to encourage selfish individuals to vaccinate optimally, vaccination should be subsidized.

Bibliographie

1. C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1) :195–259, 2009.
2. J. Doncel, N. Gast, and B. Gaujal. Mean-field games with explicit interactions. *In preparation*, 2015.
3. D. Gomes, J. Mohr, and R. Souza. Continuous time finite state mean field games. *Applied Mathematics & Optimization*, 68(1) :99–143, 2013.
4. T. G. Kurtz. *Approximation of population processes*, volume 36. SIAM, 1981.
5. L. Laguzet and G. Turinici. Individual vaccination as Nash equilibrium in a SIR model : the interplay between individual optimization and societal policies. June 2015.
6. J.-M. Lasry and P.-L. Lions. Mean field games. *Japanese Journal of Mathematics*, 2(1) :229–260, 2007.

Efficient content delivery in the presence of impatient customers and multiple content types

M. Larrañaga[1], O. J. Boxma[2], R. Núñez-Queija[3], M. S. Squillante[4*]

[1] L2S UMR CNRS 8506, CentraleSupelec
[2] Eurandom, The Netherlands
[3] CWI, Amsterdam, The Netherlands
[4] Mathematical Sciences Department, IBM, USA
maialen.larranaga@supelec.fr

1. Introduction

In this paper we investigate a system that combines batch services with abandonment of customers. This model consists of a multi-class M/M/1 multi-server queue with an adapted service process. This service process consists in delaying the customers that demand the delivery of the same content for batching of service. Delays due to batching come at the cost of abandonment. In particular, customers may abandon the system while waiting to be served (expiration of their deadlines), for which we penalize the system at a fixed cost per abandoning customer. Such penalties can either represent the loss of the customer or the cost of serving the customer on an expensive back-up service.

The characterization of an optimal control for the case in which customers require a different content type is out of reach. We therefore develop approximations to tackle the problem, in particular, we study Whittle's index.

Index Rules have enjoyed a great popularity, since a complex control problem whose solution might, a priori, depend on the entire state space turns out to have a strikingly simple structure. In a seminal work, Whittle introduced the so-called Restless Multi Armed Bandit Problems (RMABP), see [2]. In a RMABP all bandits (in our study a bandit is a group of customers that demand a specific type of content) in the system incur a cost, the scheduler selects one bandit to be made active, but all bandits might evolve over time according to a stochastic kernel that depends on whether the bandit was active or *frozen*. The objective is to determine the control policy that, based on the entire state-space description, selects the bandit with the objective of optimizing the average performance criterion. Whittle introduced an approximate control policy of index-type, which is nowadays referred as Whittle's index.

The model under study can be cast as an RMABP. We will therefore aim at deriving the Whittle index to obtain a heuristic for the original problem.

2. Model Description

We consider a multi-class M/M/1 queue with batch service, M servers with infinite service capacity and customers abandonment. Customers that demand content type $k \in \{1, \ldots, K\}$ arrive according to a Poisson process with rate λ_k and have an exponentially distributed service requirement with mean $1/\mu_k$, which is independent of the batch size. Customers that are waiting in the queue abandon after an exponentially distributed amount of time with mean $1/\theta_k$. Furthermore, all interarrival times, service requirements and abandonment times are independent.

Each server can only deliver one content type at a time. In every decision epoch the policy ϕ chooses whether to process the demands for a content or not. Once a customer has been admitted for service we assume that it can not abandon the system. Let $N_k^\phi(t) \in \{0, 1, \ldots\}$ denote the number of customers with a demand for content k that are waiting in the queue at time t under the policy ϕ. We will denote them class-k customers. Let $S_k^\phi(\vec{N}^\phi(t)) \in \{0, 1\}$ denote the decision with respect to class-k customers at time t under policy ϕ when there are $\vec{N}^\phi(t)$ customers present in the system, with $\vec{N}^\phi(t) = (N_1^\phi(t), \ldots, N_K^\phi(t))$. Namely, $S_k^\phi(\vec{N}^\phi(t)) = 0$ if the server does not serve class k, and $S_k^\phi(\vec{N}^\phi(t)) = 1$ if the server decides to take a class-k batch into service. Due to the infinite capacity of the server we assume that, as soon as the server is activated, *i.e.*, $S_k^\phi(\vec{N}^\phi)(t) = 1$, all customers that are waiting in the queue k initialize their service. Hence, the batch size upon activation equals the number of customers waiting in the queue, $N_k^\phi(t)$.

We assume that the service requirements are exponentially distributed with rate $\mu_k < \infty$. Upon activation of queue k the server takes a batch of size $N_k^\phi(t)$ into service, and allocates an exponentially distributed amount of time to process it. While the server is busy, new customers might arrive to the queue. In this case, the server is not allowed to take a new batch into service until service completion of

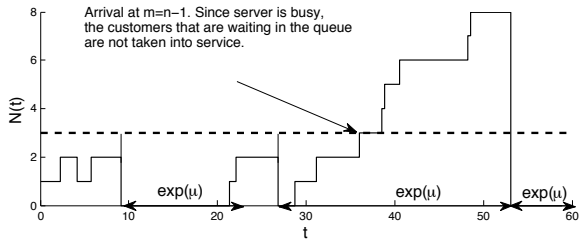

Figure 1: Simulation of process $N(t)$ under threshold $n = 3$. $exp(\mu)$ refers to the busy period of the server. $N(t)$ not only depends on n (the policy) but also on the length of each busy period.

the previous batch; see Figure 1 around $t = 37$.
Let us denote by c_k the cost per unit of time class-k customers are held in the queue, δ_k the penalty for class-k customers abandoning the queue and by c_k^s the cost per unit of time the server is busy. The objective of the present work is to find the policy ϕ so as to minimize

$$\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T [\sum_{k=1}^K \tilde{c}_k N_k^\phi(t) + c_k^s S_k^\phi(\vec{N}^\phi(t))]\mathrm{d}t\right],$$

if $\mu < \infty$, where $\tilde{c}_k = c_k + \delta_k\theta_k$. Due to ergodicity of the system, the time-average optimal policy is equivalent to the optimal policy in steady-state, and hence we want to find ϕ such that

$$\min_\phi \sum_{k=1}^K \left(\tilde{c}_k\mathbb{E}[N_k^\phi] + c_k^s\mathbb{E}(\mathbf{1}_{\{S_k^\phi(\vec{N}^\phi)=1\}})\right), \quad (1)$$

subject to $\sum_{k=1}^K S_k^\phi(\vec{N}^\phi) \le M$. The problem described above is a Markov Decision Process and obtaining an optimal solution is out of reach. In the next section we propose a well-performing heuristic.

3. Whittle's index

The approach by Whittle is based on relaxing the original problem, allowing the constraint on the servers to be satisfied on average, that is,

$$\limsup_{T\to\infty} \frac{1}{T}\int_0^T \sum_{k=1}^K S_k^\phi(\vec{N}^\phi(t))\mathrm{d}t \le M. \quad (2)$$

This allows to decompose the original control problem into individual problems for each class of customers (we therefore drop the dependency on k from now on). Whittle's index can then be interpreted as the Lagrange multiplier of the constraint such that a given state joins the passive set.

The objective is now to determine the policy that solves (1) under Constraint (2). This can be solved by considering the uni-dimensional unconstrained control problems

$$\limsup_{T\to\infty} \left(\tilde{c}\mathbb{E}[N^\phi] + (c^s + W)\mathbb{E}(\mathbf{1}_{\{S^\phi(N^\phi)=1\}})\right), \quad (3)$$

where W is the Lagrange multiplier that can be interpreted as a subsidy for passivity.
We first prove that an optimal policy that solves (3) is of threshold type, that is, there exists n such that it is optimal not to allocate the server for all states $m \le n$ and it is optimal to allocate the server otherwise. The proof can be found in [1].

Proposition 1 *There exists $n = 0, 1, 2, \ldots$ such that $S^n(m) = 0$ for all $m \le n$ and $S^n(m) = 1$ otherwise.*

Having proven threshold type of policies to be optimal, one can define Whittle's index as described in the following algorithm.

Proposition 2 *Let $\mathcal{N}_i = \mathbb{N} \cup \{0\}\backslash\{0, \ldots, n_i\}$ for a given n_i, let $P^n = \mathbb{E}(\mathbf{1}_{\{S^n(N^n)=1\}})$ and be non-increasing, and define $\beta(\cdot)$ as follows: Define $n_0 = 0$ and*
Step i. Compute

$$\beta_i := \inf_{n\in\mathcal{N}_{i-1}} \tilde{c}\frac{\mathbb{E}(N^n) - \mathbb{E}(N^{n_{i-1}})}{P^{n_{i-1}} - P^n} - c^s, i \ge 1, \quad (4)$$

and denote by n_i the largest $n \in \mathcal{N}_{i-1}$ such that (4) is minimized and define $\beta(n) = \beta_i$ for all $n_i > n \ge n_{i-1}$. If $n_i = \infty$ stop and let $\beta(n) = \beta_i$ for all $n \ge n_i$, otherwise jump to step $i + 1$.
Then, β_i is strictly non-decreasing in i. and $\beta(n)$ defines Whittle's index.

Whittle's index policy prescribes to serve the M classes of customers with highest Whittle's index. We have numerically observed that Whittle's index policy behaves close to optimal for heavy-traffic and light-traffic regimes. The latter however has not been proven analytically.

Bibliographie

1. M. Larrañaga, O.J. Boxma, R. Nunez-Queija, M.S. Squillante – Efficient content delivery in the presence of impatient jobs. – Proceedings of ITC 2015.
2. P. Whittle – Restless bandits: Activity allocation in a changing world. – Journal of Applied Probability, 25:287-298, 1988.

Admission Control with Machine Learning in Software Defined Networks

Jérémie Leguay, Lorenzo Maggi, Moez Draief,
Stefano Paris, Symeon Chouvardas,
Huawei Technologies, FRC Research Lab
(Boulogne-Billancourt, France)
Email: firstname.lastname@huawei.com

1. Admission control in SDNs

Software-Defined Networking (SDN) technologies have radically transformed network architectures. They provide programmable data planes that can be configured from a remote controller platform. This creates an opportunity to implement routing processes that are more efficient than classic ones: in fact, the controller can take real-time decisions at a (logically) centralized location using an accurate and global view of the network. Moreover, SDN controllers have a tremendous computational power compared to legacy embedded devices. This encourages the development of smarter network control planes using cutting-edge optimization and machine learning techniques. A key task of the SDN controller is the Admission Control on incoming connection requests, in an online manner. Its goal is to gracefully manage service requests when the network becomes highly utilized, as new incoming requests arrive. It accepts or drops new requests depending on the resource availability. Non-myopic decisions have to be made to maximize a given profit, such as the total accepted throughput, the financial revenue generated or the quality of service experienced by users. Admission Control can be formulated as an online packing problem, as the goal is to maximize the number of accepted requests (subject to feasibility constraints). The challenge in this context comes from the online nature of the optimization problem. New variables and additional constraints are revealed sequentially, as soon as an arrival or departure of a flow occurs in the system. The theory on online algorithms has evolved significantly in the last decade for this type of problems. Algorithms with guaranteed over the offline optimal, which has full information on the future state of the system, have been proposed for online packing problems. For these guarantees to hold, admission decisions need to be consistently taken. The centralized nature of SDN lies the ground to apply online algorithms for admission control.

2. Online Algorithms

Motivated by the considerations above, we first review and adapt, well-known and recent algorithms from the online literature to the admission control problem in SDN. We then test their performance under different traffic conditions to understand and highlight the strengths and weaknesses of each of them. Traditionally, online algorithms for admission control fall into two main categories: i) worst-case and ii) average-case. Worst-case algorithms are characterized by max-min performance guarantees under specific worst-case scenarios where a malicious adversary chooses the worst possible sequence of connection requests. Due to their conservative nature, they generally underperform under more standard traffic conditions. One of the first online algorithms for admission control has been AAP [3]. The algorithm is $O(\log n)$-competitive, meaning that it cannot reject $O(\log n)$ times more requests than the offline optimal (n being the number of nodes). Buchbinder et al. proposed in [2] a framework to derive algorithms for online packing and covering problems with performance guarantees in the worst-case scenario. Such framework developed the theory behind the initial intuition of AAP. On the other hand, ii) average-case algorithms show high expected performance over random traffic conditions, but cannot guarantee good performance in specific adversarial scenarios. The Primal-Beats-Dual (PBD) algorithm has been introduced by Kesselheim et al. in [3]. It computes the optimal (fractional) solution of the relaxed Linear Program (LP), by considering all the past requests and scaling the capacity of the graph. It then attempts to route the request over a path randomly selected, with a probability proportional to the value of the computed fractional solution. PBD suffers from computation complexity. Agrawal et al. [4] have proposed a fast algorithm with multiplicative updates to solve this issue. It applies to i.i.d. and random order inputs. It implements an efficient stochastic gradient descent that we adapted for our admission control problem.

3. Admission Control with Experts

As observed in practice, worst-case and average-case online algorithms for admission control are better than the naive and greedy strategy which accepts every demand until bottlenecks appears.

However, there is no algorithm outperforming all the remaining ones under all traffic conditions. Luckily, modern SDN control platforms, which are running on top of commodity servers and are built upon cutting-edge distributed computing technologies, enable the parallel execution of different algorithms to solve a single decision problem. Such algorithmic architecture, called boosting or prediction with expert advice setting [6], is commonly used in machine learning. It executes all the algorithms in parallel and attempts to track the best one in an online fashion. The bulk of the literature on experts focuses on proving theoretical performance bounds in the basic non-reactive scenario where the action taken by the decision maker does not affect the state of the system, which is definitely not the case for admission control. We identified that the Strategic Expert meta-Algorithm (SEA) [5] applies to out reactive setting. Under some stationary conditions on the system (in our case, on the traffic load on the links), SEA is proven to perform at least as well as the oracle which steadily selects the algorithm with best average performance. We finally show that this approach achieves very good performance in practice, and is able to successfully overcome the several limitations imposed by the inherent online and hard-to-predict nature of the problem at hand. We evaluate the performance of the online algorithms under realistic conditions, using the real-life dataset captured in 2006 by Uhlig et al. [7] on the GEANT network.

Bibliography

1. B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line routing"', in Proc. FOCS, 1993.
2. N. Buchbinder and J. Naor, "Improved bounds for online routing and packing via a primal-dual approach", in Proc. FOCS, 2006.
3. T. Kesselheim, A. Tönnis, K. Radke, and B. Vöcking, "Primal beats dual on online packing LPs in the random-order model", in Proc. ACM STOC, 2014.
4. S. Agrawal and N. R. Devanur, "Fast algorithms for online stochastic convex programming", in Proc. ACM SODA, 2015.
5. D. P. de Farias and N. Megiddo, "How to combine expert (and novice) advice when actions impact the environment?", in Advances in Neural Information Processing Systems, 2003.
6. N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press, 2006.
7. S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community", ACM SIGCOMM Computer Communication Rev, vol. 36, no. 1, 2006.

Blocking Evaluation of dynamic WDM networks without wavelength conversion

Nicolás Jara[1,2], Reinaldo Vallejos[2],
Gerardo Rubino[1]

Affiliation
[1]INRIA Rennes - Bretagne Atlantique, Rennes, France
[2]Universidad Técnica Federico Santa Maria, Valparaíso, Chile
nicolas.jara@usm.cl, reinaldo.vallejos@usm.cl, gerardo.rubino@inria.fr

1. Introduction

The rapid increase on demand for bandwidth from existing networks has caused a growth in the use of technologies based on WDM optical infrastructures [4]. Currently, this type of network is operated statically [4], i.e., the route assigned to any user is permanently assigned from source to destination. However, this type of operation is inefficient in the usage of network resources, especially for low traffic loads, which is the most common case.

One way to help overcome the inefficiencies of static networks is to migrate them to networks working dynamically. This operation mode consists in allocating the resources required only when the user has data to transmit. A possible lack of resources to successfully transmit can happen because dynamic networks are designed to save costs with the less possible amount of resources and also to be efficient avoiding burst losses (blocking). To achieve a balance between these two aspects, the network must be designed such that the connection blocking probability is less than or equal to a design parameter B. The evaluation of this metric allows to determine whether or not each network user (each connection) is being treated with the required quality of service.

Another technology useful to improve this static network operation is the optical switches wavelength conversion capacity. In the specialized literature the architecture of dynamic WDM optical networks without wavelength conversion is considered the next generation of optical networks, due the dynamic resource assignment (who origins optical networks) already exist and the wavelength conversion capacity is still in an embryonic phase.

Usually the blocking probability is evaluated through simulation [3, 5]. However, this technique is in general very slow compared with the solution obtained via an appropriate mathematical method. The evaluation speed is relevant, because when solving problems of higher order (e.g. routing or fault tolerant mechanism), it is necessary to calculate this metric several times. Thus, a mathematical computational method is required fast and efficient. However, this is difficult due to important aspects to take into account such as traffic load, wavelengths capacity and continuity, network topology, etc. Several models have been proposed to evaluate the blocking probability [2, 1].

In this document we propose a new approach to evaluate the blocking probability (burst losses) in Dynamics WDM optical networks without considering wavelength conversion.

Network and Traffic model

The network is represented by graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of network nodes , and $\mathcal{L}$ is the set of unidirectional links, with respective cardinalities N and L. The set of connections $\mathcal{X} \subseteq \mathcal{N}^2$, with cardinality X, is composed of all the source-destination pairs with communication between them.

To represent the traffic between a given source-destination pair an ON-OFF model is used. During the ON period, with average length t_{ON}, the source transmits at a constant transition rate. During the OFF period, with average length t_{OFF}, the source refrains from transmitting data. Consequently, the traffic load for each individual connection ρ, is given by the following expression:

$$\rho = \frac{t_{ON}}{t_{ON} + t_{OFF}}. \quad (1)$$

Blocking evaluation strategy

Let $\mathcal{R} = \{r_c \mid c \in \mathcal{X}\}$ be the set of routes that enable communication among the different users (connections) in the network, where r_c is the route associated with connection $c \in \mathcal{X}$. For every link $\ell \in \mathcal{L}$, we denote by W_ℓ the number of wavelengths associated with link $\ell \in \mathcal{L}$.

Given the complexity of the exact evaluation of the blocking probability considering the aspects explained before, we developed a strategy to obtain an accurate while light cost computational

scheme. Note that the most important aspect to consider is the wavelength continuity problem, because there is not wavelength conversion capability. This means that when a connection transmits, it must use the same wavelength on each link that belongs to its route. We explain below the different steps of this procedure.

- First, the network $\mathcal{G}$ with W_ℓ wavelengths on link ℓ, is divided into a sequence of W_ℓ networks denoted $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_{W_\ell}$, where each link capacity is equal to 1. This will avoid the problem of wavelength continuity due to the fact that there is only 1 wavelength per network. However, to emulate the operation of the this optical networks, exists a sequential interaction between this networks $\mathcal{G}_w$. This means, all the blocked connection request rate in the $\mathcal{G}_w$ network will be considered as the connection request rate on the $\mathcal{G}_{w+1}$ network. To consider the interaction between the W_ℓ networks, we consider a dependency between $t_{ON_{c,w}}$ and $t_{OFF_{c,w}}$, the mean ON and OFF periods "seen" on any network $\mathcal{G}_w$ by all the connections $c \in \mathcal{X}$. These values will be explained below.

- Next, an analytical model to evaluate the blocking probability of each network of the sequence is developed. The blocking probability of the $\mathcal{G}_w$ network, which we denote by BC_c^w, is evaluated assuming that the states of the links that constitute route r_c are independent, that is,

$$BC_c^w = 1 - \prod_{\ell \in r_c} (1 - BL_{\ell,w}^c) \tag{2}$$

  where $BL_{\ell,w}^c$ is the blocking probability of connection c on link ℓ with $\ell \in r_c$ in the network $\mathcal{G}_w$. $BL_{\ell,w}^c$ is evaluated as follows:

$$BL_{\ell,w}^c = \frac{1 - \pi_0^{\ell,w} - \pi_c^{\ell,w}}{1 - \pi_c^{\ell,w}}, \tag{3}$$

  where $\pi_c^{\ell,w}$ is the probability that connection c is using link ℓ in network $\mathcal{G}_w$, and $\pi_0^{\ell,w}$ is the probability that no connection is using the link ℓ in network $\mathcal{G}_w$. These probabilities are calculated by means of a Markov Chain analysis considering the $t_{ON_{c,w}}$ and $t_{OFF_{c,w}}$ values, not shown here for lack of space.

- The interaction between the networks in the sequence is considered in the $t_{ON_{c,w}}$ and $t_{OFF_{c,w}}$ values of network $\mathcal{G}_w$. The parameter $t_{ON_{c,w}}$ does not depend of $\mathcal{G}_w$, because it is the time used by the source to transmit, i.e. $t_{ON_{c,w}} = t_{ON_c}$, for each network $\mathcal{G}_w$. To represent the dependencies between the W_ℓ networks, the $t_{ON_{c,w}}$ value must change with w. This changes carries 3 types of dependencies.
  - Bottom-top: All non blocked connections in the network $\mathcal{G}_w$ make $t_{OFF_{c,w+1}}$ grow by 1 cycle.
  - Top-Bottom: All blocked connections in network $\mathcal{G}_1$, but non blocked in network $\mathcal{G}_{m+1}$, with $1 \leq m \leq W_\ell - 1$, make $t_{OFF_{c,1}}$ grow by 1 cycle.
  - General blocking: All blocked connections in the final network $\mathcal{G}_{W_\ell}$ make the first network increase by t_{OFF}.

  Then, to consider these dependencies, a fixed point procedure is proposed.

- Finally, the average network blocking probability of a dynamic optical network, B_{net}, is evaluated as follows:

$$B_{net} = \frac{\sum_{c \in \mathcal{X}} \lambda \cdot BC_c}{\sum_{c \in \mathcal{X}} \lambda}, \tag{4}$$

  where BC_c is the general connection blocking probability, calculated by $BC_c = \prod_{\text{all } w} BC_c^w$.

Bibliographie

1. V. Abramov, Shuo Li, Meiqian Wang, E.W.M. Wong, and M. Zukerman. Computation of blocking probability for large circuit switched networks. *Communications Letters, IEEE*, 16(11):1892–1895, November 2012.
2. Luiz H. Bonani and Iguatemi E. Fonseca. Estimating the blocking probability in wavelength-routed optical networks. *Optical Switching and Networking*, 10(4):430 – 438, 2013.
3. Biswanath Mukherjee. *Optical WDM networks*. Springer Science & Business Media, 2006.
4. A. A M Saleh and Jane M. Simmons. Technology and architecture to enable the explosive growth of the internet. *Communications Magazine, IEEE*, 49(1):126–132, January 2011.
5. A. Zapata-Beghelli and P. Bayvel. Dynamic versus static wavelength-routed optical networks. *Lightwave Technology, Journal of*, 26(20):3403–3415, Oct 2008.

Backward-Shifted Coding (BSC) for HTTP Adaptive Streaming

Zakaria Ye*, Rachid EL Azouzi*, Tania Jimenez*, Stefan Valentin+

University of Avignon, France*, Huawei Technologies, France+
{ zakaria.ye, rachid.elazouzi, tania.jimenez}@univ-avignon.fr, stefan.valentin@huawei.com

1. Backward-Shifted Coding

Before introducing our Backward-Shifted Coding, we describe the SVC [1], which composed of a H264/AVC-compatible base layer and one or more enhancement layers. These layers increase the SNR fidelity, spatial and/or temporal quality of the video when added to the base layer. In fact, the description of our scheme will be based on the H.264/SVC codec.
The BSC scheme is entirely client driven. The main idea of this scheme is to decompose the segment or the GoP in base layer frames and enhancement layer frames and shift them. The lag between the base layer and the enhancement layer frames is ϕ. The base layer frames are less quality and they are sent before the enhancement layer frames. Each frame n has it enhancement layer in subsequent frame $n + \phi - 1$ (see fig. 1). Thus, when the enhancement layer is missed, the player can still playout the base layer frame.
In our BSC scheme, each segment or block is encoded with different rates based on the network conditions and the scheduling technique. Typically, that means the encoding rate of the base layer and the encoding rate of the combined base and enhancement layers. The encoding rate of the base layer may differ from one segment to another. That is an interesting property of the BSC system to be explain later. At the user side, incoming bits are reassembled into video frames by the decoder.
Under DASH, each video client is divided into multiple small segments at the media server. Various representations of a segment or block are proposed by using different encoding rates. Hence, the BSC scheme can be used with evident client using the DASH framework. The base layer is re-encoded for each segment, possibly with a different encoding rate, depending on the network throughput measurements. It is a runtime encoding process like in live video streaming systems. Therefore, with BSC under DASH, the adaptation engine requests the two appropriate segments with different bitrates (base layer and enhancement layers).

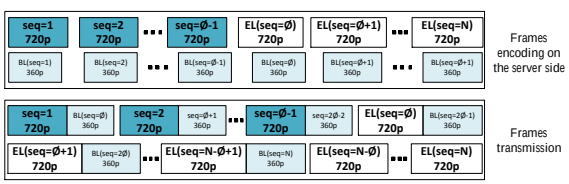

FIGURE 1 – Using SVC in Backward-Shifted Coding

2. Bitrate Adaptation with Backward-Shifted Coding

2.1. System Description

We consider a video streaming system using the Backward-Shifted Coding described. The server holds the media segments and a HTTP server. The client holds an adaptation engine and a playout buffer where the video frames are decoded and stored to be display on the screen. The main goal of the adaptation engine is to estimate the throughput and select the bitrates of the next segment to be downloaded.
We assume N to be the size of the video file in number of frames. Let d be the buffer playout frequency (e.g., 30fps). The video duration Δ (in seconds) is N/d. A video file is a set of consecutive video segments or chunks, $\nu = \{1, 2, ..., K\}$, each of which contains L seconds of video and encoded with different bitrates. The number of segments K in the video is Δ/L. We assume $\mathcal{R} = \{R_1, R_2, ..., R_r\}$ to be the set of available bitrates. In BSC system, the player downloads the video segment k with bitrate $(R_{ki}, R_{kj}) \in \mathcal{R}$ where R_{ki} is the bitrate of the basic layer and R_{kj} is the bitrate of the combined basic layer and enhancement layer.

2.2. Adaptation methods in BSC

The goal of the bitrate adaptation is to maximize the quality of experience of the user. We consider the throughput based method (TBA) (current segment throughput) and the buffer based method (BBA) (buffer level) algorithms. For TBA, the algorithm selects the adequate bitrates after the download of the current segment. The pseudo-code is provided in algorithm 1 where $R_<(t_i)$, $R_>(t_i)$, R_{min} and R_{max} are such that $R_{min} < ... < R_<(t_i) < \hat{A}(t_i) < R_>(t_i) < ... < R_{max}$.

Algorithm 1: Adaptation Algorithm

Input:
$\hat{A}(t_i)$: the estimated throughput of the previous segment i
$R_i(BL)$: the bitrate of the base layer of the previous segment i
Output:
$R_{i+1}(BL)$: the bitrate of the base layer of the next segment $i+1$
$R_{i+1}(BL+EL)$: the bitrate of the combined layers of the next segment $i+1$

```
R_<(t_i) ← Â(t_i)^↓ ;
R_>(t_i) ← Â(t_i)^↑ ;
if Â(t_i) ≤ R_min then
    R_{i+1}(BL) := R_{i+1}(BL+EL) := R_min ;
else
    if Â(t_i) ≥ R_max then
        R_{i+1}(BL) := R_{i+1}(BL+EL) := R_max ;
    else
        if Â(t_i) < R_i(BL) then
            R_{i+1}(BL) := R_<(t_i) ;
            R_{i+1}(BL+EL) := R_>(t_i) ;
        else
            R_{i+1}(BL) := R_i(BL)^↑ ;
            R_{i+1}(BL+EL) := R_{i+1}(BL)^↑ ;
```

For BBA we define three buffer thresholds B_{min}, B_{low} and B_{high} and make the bitrate selection decision on the buffer filling level. Algorithm 1 is used at the beginning of this method to increase quickly the bitrate at the beginning of the video playback in order to maximize the video quality.

3. Simulations and Numerical results

3.1. Simulation setup

We use MATLAB to simulate the event-driven bitrate switching system. Our traffic model (see fig.

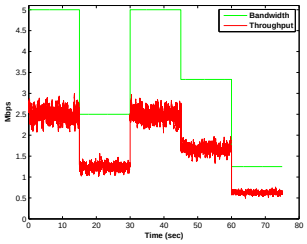

FIGURE 2 – Background traffic with Level shift

2) simulates a realistic network bandwidth variations with congestion level shift where they inject background TCP traffic between the server and the client. The link capacity between the server and the client is set to be 5Mbps. The traffic rate jumps between four different levels, oscillations within the same level are due to TCP congestion control mechanisms. The link capacity is higher than the TCP throughput.

3.2. Numerical Results

This set of experiments compares the requested bitrate with BSC and DASH under the network conditions of fig. 2. The file size in the experiments is up to 100 seconds of video while the playabck frequency is 25 fps. We consider the following set of available bitrates $\{140, 250, 420, 760, 1000, 1500, 2100, 2900\}(Kbps)$.
The video segment duration is set to 2 seconds.
Figure 3 shows the requested bitrate for BSC and

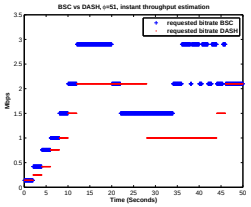

FIGURE 3 – BSC vs DASH instant throughput estimation

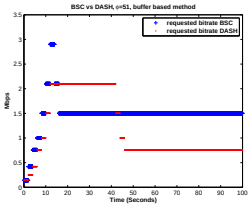

FIGURE 4 – BSC vs DASH buffer based method

DASH for TBA where the throughput is estimated over 1 segment. The average video bitrate is much higher with BSC system for a few additional bitrate swithcings. These bitrate switchings are tolerable since they are between two consecutive bitrates. We can still cope these switchings by forcing the system to stay at the base layer frames if the time spent at the base level does not exceed a certain threshold.
Figure 4 shows the requested bitrate for BSC and DASH for BBA $B_{min} = 5sec$, $B_{low} = 7sec$, $B_{high} = 50sec$. BSC system achieves a better video quality and decreases the bitrate switchings compared to TBA.

4. Conlusion

We proposed a novel coding scheme to improve the performance of the HTTP adaptive video streaming. BSC is inspired from the FEC method. The media server transmits two shifted segments with different bitrates, i.e., a base layer with enhancement layers, in order to enhance the video quality. We proposed bitrate adaptation algorithms in BSC. We further performed simulations to show the efficiency of BSC system compared to existing DASH solutions.

Bibliographie

1. ISO/IEC – Coding of Audio-visual Objects - Part 10 : Advanced Video Coding 14496-10 :2012 Information Technology, 2012.

Resource Allocation for in-Network Computations

Apostolos Destounis[1], Georgios S. Paschos[1], Iordanis Koutsopoulos[2]

[1]Mathematical and Algorithmic Sciences Lab, Huawei technologies FRC
20, quai du Point du Jour
92100 Boulogne-Billancourt, France
firstname.lastname@huawei.com

[2]Department of Informatics, Athens University of Economics and Business (AUEB)
76, Patission Str., 104 34, Athens, Greece
jordan@aueb.gr

1. Introduction

In this work, we ask the following question : Given a network graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with links of limited communication bandwidth and nodes of limited computation resources, what are the *performance limits* of in-network computation throughput ? Namely, what is the *maximum rate* with which computation results can be conveyed to the destination when computations take place in the network ?
We assume there exist two source nodes $s_1, s_2 \in \mathcal{N}$ and a destination node d. Edge $(m, l) \in \mathcal{E}$ between nodes m and l has a fixed capacity of R_{ml} packets per slot. A network example is given in Fig. 1.
We study a *stream* of queries, where each query concerns the computation of the sum of a datum from source 1 and a datum from source 2, while the network is agnostic to specificities of data. Upon arrival of each query, a corresponding packet (datum) is generated at each of the two source nodes, and both packets are given the same *tag*. These packets need to be summed somewhere in the network, and the result needs to be delivered to the destination d. Time is slotted, and at each slot t there are $A(t)$ newly arrived queries belonging to the same stream, random with $\mathbb{E}[A(t)] = \lambda$.
Combination of packets corresponding to a query may take place in one among a subset of nodes, denoted by $\mathcal{N}_C = \{n_1, n_2, ..., n_{N_C}\} \subseteq \mathcal{V}$; these are referred to as the *computation nodes*. Node n_i has computational capacity of C_{n_i}, measured in number of produced processed packets per slot, where each processed packet concerns the sum of two raw packets with the same tag when both are available to the computation node.

The objective of this work is to characterize the maximum rate of queries that can be accommodated by the network (referred to as the computation capacity of the network), and provide an online algorithm to achieve this capacity. We restrict ourselves to policies that use only packet routing (i.e. no network coding).

2. Queueing Structure

To capture all packet classes in the network we define the following queues (A calligraphic sign denotes a set with the tags of the corresponding packets, normal sign denotes the cardinality of this set) :

— $\mathcal{Q}_k^{(i,n)}(t), i = 1, 2$: Data queue at node k containing raw packets generated at node s_i that have to be computed at node n.

— $\mathcal{X}_n^{(i)}(t), i = 1, 2$: Computation queue at node n containing raw packets generated at node s_i that have to be computed at *this node*.

— $\mathcal{Q}_k^{(0,n)}(t), i = 1, 2$: Data queue at node k containing processed packets from computing node n, that have to be delivered at the destination node.

In addition, each computation node has queues $\mathcal{Y}_n(t)$ keeping the results of computations (see also Fig. 2) and virtual queues $H_n(t)$ tracking the computation capacity budget.
Moving packets between queues corresponds to control decisions to be taken each slot :

— The set of raw packets originated from node s_i, destined to computation node n, to be transmitted from node m to node k.

— The pairs of raw packets to be combined at each computation node n

— The set of processed packets, combined at node n, to be transmitted from node m to node k.

We have the following constraints : (i)The total number of transmitted packets over a link (kl) are limited by link capacity R_{kl} (ii) the number of combined pairs cannot exceed the computation capacity or any of the individual raw packet queue lengths, and (iii) a pair of packets can be combined only if both packets with the same tag have already arrived at the computation node.

3. Upper Bound on the Network Computation Capacity

We define a set $\widetilde{\mathcal{C}}_3$ with $3N_C$ unicast commodities, as follows : there are three commodities for each

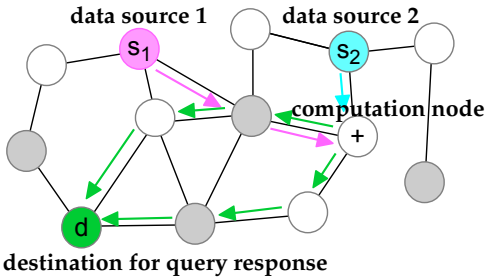

FIGURE 1 – Illustration of network computations. Shaded nodes are forwarding ones, i.e. without computation capabilities, and white nodes have computation capabilities. Arrows denote routing of raw and processed data.

computation node $n \in \mathcal{N}_C$; $(1,n)$ delivering packets from s_1 to n, $(2,n)$ delivering packets from s_2 to n, and (n,d) delivering combined packets at (computation) node n to the destination. Let $\Lambda(\mathcal{G})$ be the feasible rate region for these commodities on network $\mathcal{G}$. We can characterize the computation capacity as follows :

Theorem 1 *An upper bound on the computation capacity is given by the following optimization problem :*

$$\lambda^* = \max_{(\lambda_n)} \sum_{n\in\mathcal{N}_C} \lambda_n \quad (1)$$

$$\text{s.t.} \quad 0 \le \lambda_n \le C_n, \forall n \in \mathcal{N}_C \quad (2)$$

$$(\lambda_1,\lambda_1,\lambda_1,...,\lambda_{N_C},\lambda_{N_C},\lambda_{N_C}) \in \Lambda(\mathcal{G}) \quad (3)$$

4. Algorithm

The dynamic policy we consider here is the following :

1. **Load Balancing :** At each slot, choose $n^*(t)$ equal to

$$\arg\min_{n\in\mathcal{N}_C} \left[(1+\epsilon_B)Q_n^{(0,n)}(t) + \sum_{i=1,2} Q_i^{(i,n)}(t) + H_n(t)\right]$$

where $\epsilon_B \in (0,1)$ is a control parameter. Then, all newly arrived queries are assigned to the class that corresponds to this computation node.

2. **Routing and scheduling :** Use Backpressure over class pairs. For every link $(m,k) \in \mathcal{E}$ choose the class pair

$$\left(i^*_{mk}(t), n^*_{mk}(t)\right) = \arg\max_{i\in\{0,1,2\}n\in\mathcal{N}_C} \left|Q_m^{(i,n)}(t) - Q_k^{(i,n)}(t)\right|.$$

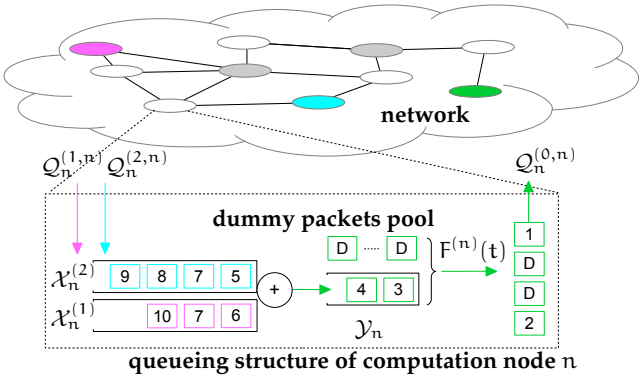

FIGURE 2 – Illustration of queueing structure for computation node n. Numbered packets are either raw or processed (green) useful packets, while packets noted with "D" are dummy packets.

Then use the capacity of the link to route (any) packets of the above class pair, from the biggest to smallest corresponding queue.

3. **Computation :** At every node $n \in \mathcal{N}_C$, all possible computations are done. If there are more pairs than the computation capacity of this node, then C_n pairs are selected using any tie breaking rule.
4. **Randomization with dummy packets :** $F^{(n)}(t) = \mathbb{1}_{\{n=n^*(t)\}}A(t)\left(1+B^{(n)}(t)\right)$ packets resulting from a computation are pushed to queue $\mathcal{Q}_n^{(0)}(t)$, where $B^{(n)}(t)$ are an i.i.d. Bernoulli random variables with mean ϵ_B. If there are not enough processed packets available at queue $\mathcal{Y}_n$, dummy packets are used.
5. **Update Virtual Queues :**

$$H_n(t+1) = [H_n(t) - C_n]^+ + \mathbb{1}_{\{n=n^*(t)\}}F^{(n)}(t).$$

The main result of this work is the performance of the online algorithm :

Theorem 2 *The online policy satisfies any query rate* $\lambda < \left(1 - \frac{\epsilon_B}{1+\epsilon_B}\right)\lambda^*$.

Theorem 2 implies that the online algorithm achieves a computation rate arbitrarily close to the upper bound.

Anticipating Resource Management and QoE Provisioning for Video Streaming

Imen TRIKI, Rachid ELAZOUZI, Majed HADDAD

University of Avignon France
imen.triki/rachid.elazouzi/majed.haddad@univ-avignon.fr

1. Resume

This study provides important insights in the design and optimization of adaptive video streaming by exploiting the knowledge of future capacity variations. We develop a framework that allows to extend the model developed in [1] by balancing system utilization and adaptive video quality. Our main contributions can be summarized as follows

i) We provide a general optimization framework for stored adaptive video delivery that accounts for operators' and clients' preferences
ii) Under the constraint of no rebuffering events, we formally obtained the optimal solution where the transmission schedule is of a threshold type and the coding strategy is of an ascending type
iii) We propose an efficient mechanism that performs close to the optimal solution, and we evaluate its performance and robustness using realistic traces.

2. Problem Formulation

We propose an optimization model in which we minimize an objective function $\mathcal{F}$ in respect of some constraints. The trend of this function depends on a fixed parameter a that adjusts the trade-off between the network utilization cost and the user QoE.

$$\min_{(r,\gamma)} \mathcal{F}(r,\gamma) = \frac{1}{T}\int_0^T \frac{r(t)}{c(t)}dt - a\frac{\sum_{j=1}^{j=L} w_j \int_0^T \gamma_j(t) r(t) dt}{S_L} \tag{1}$$

$$s.t \begin{cases} \int_0^t \frac{\lambda\, c(t)\gamma_1}{b_L} \geq l(t) & \forall t \leq T \\ \int_0^t \sum_{j=1}^{j=L} \frac{\lambda\, r(t)\gamma_j(t)}{b_L} \geq l(t) & \forall t \leq T \\ \int_0^T \sum_{j=1}^{j=L} \frac{\lambda\, r(t)\gamma_j(t)}{b_L} = l(T) \end{cases}$$

where :
c : characterizes the network available capacity
r : characterizes the network transmission schedule
b_i designs bitrate level i ; $b_i \leq b_j$ for $i \leq j$
γ_j : characterizes bitrate level j
w_j : a score assigned to bitrate level j
T : the video length in s
S_L : the video size in bits with the highest quality
λ : the video speed lecture
l : the constraint function of the buffer state evolution

3. Problem Resolution

3.1. Threshold strategy

Definition 1. *Giving the network capacity c, we define the threshold transmission schedule by*

$$r_{th}(t) = \begin{cases} c(t) & \text{if } c(t) \geq \alpha \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

Proposition 1. *Assume that there exists a feasible solution that satisfies the constraints in (1), then there exists an optimal strategy $(r_{th}, \gamma_{r_{th}})$ of optimisation problem (1), where r_{th} is a threshold transmission schedule.*

3.2. Ascending coding rate approach

Definition 2. *We say a bitrate level strategy is* **ascending** *if the quality level of segments increases during the session, i.e., for all $0 \leq t \leq t' \leq T$*

$$b(t) \leq b(t') \textit{ i.e., } \gamma(t) \geq \gamma(t')$$

Proposition 2. *Assume that there exists a threshold-based solution (r_{th}, γ) that satisfies constraints in (1), then there exists a threshold-based ascending bitrate level solution (r'_{th}, γ') that optimizes problem in (1).*

4. Algorithmic approaches

4.1. Principal of optimal solution

(i) We first look at all the possible values of $\alpha \in [\alpha_{min}, \alpha_{max}]$ that satisfy the constraints in (1) while associating to each one the highest possible video quality, (ii) Suppose that we obtain a set of M possible thresholds $\{\alpha_i, i = 1, 2, \ldots, M;\ \alpha_i < \alpha_j\ ,\ i < j\}$. Therefore, for each threshold and its corresponding video quality, we compute the resulting cost function $\mathcal{F}$, (iii) The optimal solution corresponds to the one that minimizes $\mathcal{F}$. The accuracy of this algorithm increases with M at the expand of increasing complexity.

4.2. Heuristic for a near-optimal solution

Let γ_α and $\mathcal{F}_\alpha$ be, respectively, the ascending bitrate level strategy and the cost function under r_α-based transmission schedule. The main steps of the

proposed heuristic are described in Algorithm 1, where INVEST represents the approach for generating sub-optimal thresholds and AWARE represents the heuristic for setting sub-optimal ascending bitrate levels.

Algorithm 1: Heuristic for a near-optimal solution

Data: c, VideoProperties, L, w, Q

$\alpha \leftarrow c_{min}$; $i \leftarrow 1$;

$[\text{PossibleTransmission}, r_\alpha, \gamma_\alpha] =$ ***AWARE***$(c, \alpha, \text{videoProperties}, L)$

while PossibleTransmission **do**

  $\mathcal{F}_\alpha = \text{computeObjFunction}(c, r_\alpha, \gamma_\alpha, w)$

  $i = i + 1$

  $\alpha =$ ***INVEST***(c, i, Q)

  $[\text{PossibleTransmission}, r_\alpha, \gamma_\alpha] =$ ***AWARE***$(c, \alpha, \text{videoProperties}, L)$

end

$\mathcal{F}^*_{\alpha^*} = \min\{\mathcal{F}_\alpha\}$

$\alpha_{th} = \alpha^*$

return $(\alpha_{th}, \gamma_{\alpha_{th}})$

4.2.1. Heuristic for generating thresholds : INcrease with VariablE foot STep (INVEST)

The increase on α is performed by adding a variable footstep at each iteration depending on the dynamic of the network capacity. We fix the amount of data that we wish to abandon at each step (denoted by Q). Then, we adjust the value of α which gives the corresponding variable footstep.

4.2.2. Heuristic for Anticipating qoe With Ascending bitRate lEvels (AWARE)

We start by assigning the lowest level to all video segments. Then, we keep increasing the levels progressively starting by the end of the video as long as the constraints are satisfied. Once a constraint is violated, we choose the previous level of the segment. The number of loops is equal to $L - 1$. To reduce at maximum the startup delay, we set by default the buffering-cache segments to the lowest video quality and use a greedy [1] transmission instead of using a threshold-based transmission. An inherent advantage of this algorithm is that it ensures a progressive increase of the quality levels instead of an aggressive increase as in the optimal solution, which is quite more appreciable for the user's perception.

1. A greedy transmission uses all the available network capacities.

5. Results

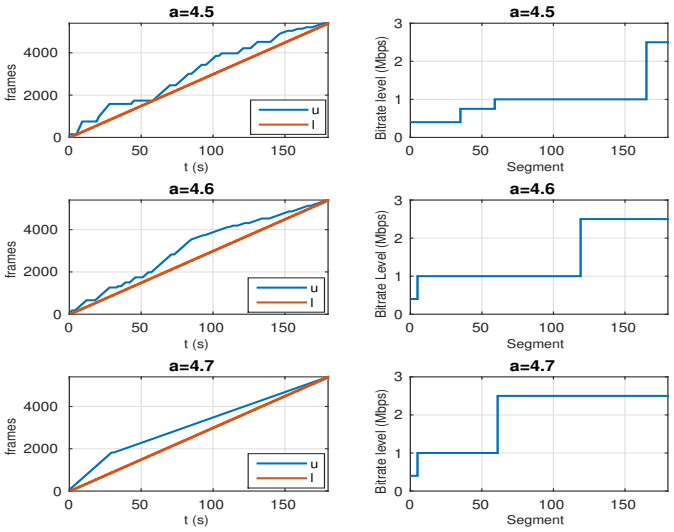

FIGURE 1 – Playback buffer state evolution and corresponding bitrate levels for different a.

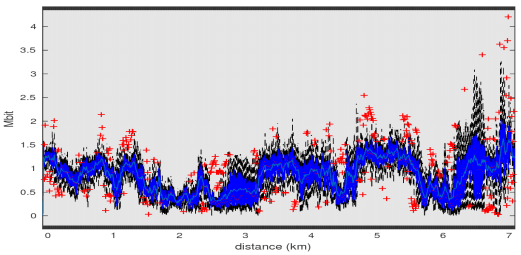

FIGURE 2 – Experimental real spacial variations of the capacity for the tramway Ljabru-Jernbanetorget trajectory.

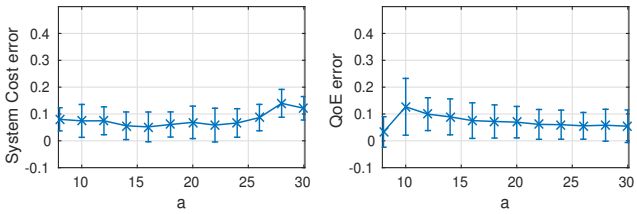

FIGURE 3 – Average error rate on the system performance under throughput prediction errors.

1. Lu, Zheng and de Veciana, Gustavo. – Optimizing stored video delivery for mobile networks : The value of knowing the future. – INFOCOM , 2013.

Liste des participants

- Farah Ait Salaht, LIP6 Paris
- Ahmad Al Sheikh, QoS Design Toulouse
- Urtzi Ayesta, LAAS Toulouse
- André-Luc Beylot, ENSEEIHT Toulouse
- Mohamed El Hedi Boussada, MEDIATRON Tunis
- Olivier Brun, LAAS-CNRS Toulouse
- Hind Castel, SAMOVAR Evry
- Apostolos Destounis, Huawei France Research Lab Boulogne Billancourt
- Josu Doncel, INRIA Grenoble
- Stephane Durand, INRIA Montbonnot
- Rachid Elazouzi, Laboratoire Informatique d'Avignon
- Samer El Zant, IRIT, Toulouse
- Jean-Michel Fourneau, Université de Versailles Saint Quentin
- Jean-Marie Garcia, LAAS Toulouse
- Nicolas Gast, INRIA Grenoble
- Bruno Gaujal, INRIA Grenoble
- Lennart Gulikers, INRIA Palaiseau
- Nidhi Hegde, Nokia Paris
- Nicolas Jara, INRIA Rennes
- Alain Jean-Marie, Inria Montpellier
- Baptiste Jonglez, ENS de Lyon
- Maialen Larrañaga, Centrale-Supélec Gif-sur-Yvette
- Patrick Loiseau, EURECOM Sophia-Antipolis
- Xiaoyan Ma, ENSEEIHT/IRIT TOULOUSE
- Lorenzo Maggi, Huawei France Research Lab Boulogne Billancourt
- Cristian Maxim, Inria Paris
- Yves Mocquard, IRISA Rennes
- Thi Thu Hang Nguyen, LAAS-CNRS Toulouse
- Rudesindo Núñez-Queija, CWI et Univ. van Amsterdam
- Balakrishna Prabhu, LAAS-CNRS Toulouse
- Florian Simatos, ISAE SUPAERO Toulouse
- Imen Triki, Laboratoire Informatique d'Avignon
- Maaike Verloop, IRIT/ENSEEIHT Toulouse
- Jean-Marc Vincent, LIG Grenoble
- Régnié Gwénaël, Université Paul Sabatier Toulouse
- Zakaria Ye, Laboratoire Informatique d'Avignon

Liste des contributeurs