



**HAL**  
open science

## Robust and reactive traffic engineering for dynamic traffic demands

Pedro Casas Hernandez, Lionel Fillatre, Sandrine Vaton

► **To cite this version:**

Pedro Casas Hernandez, Lionel Fillatre, Sandrine Vaton. Robust and reactive traffic engineering for dynamic traffic demands. NGI 2008: 4th EuroNGI Conference on Next Generation Internet Networks, May 2008, Krakow, Poland. pp.69-76, 10.1109/NGI.2008.16 . hal-00540898

**HAL Id: hal-00540898**

**<https://hal.science/hal-00540898>**

Submitted on 29 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Robust and Reactive Traffic Engineering for Dynamic Traffic Demands

Pedro Casas  
ENST Bretagne  
Brest, France  
Email: pedro.casas@enst-bretagne.fr

Lionel Fillatre  
UTT  
Troyes, France  
Email: lionel.fillatre@utt.fr

Sandrine Vaton  
ENST Bretagne  
Brest, France  
Email: sandrine.vaton@enst-bretagne.fr

**Abstract**—Traffic Engineering (TE) has become a challenging mechanism for network management and resources optimization due to uncertain and difficult to predict traffic patterns. Recent works have proposed robust optimization techniques to cope with uncertain traffic, computing a stable routing configuration that is immune to demand variations within certain uncertainty set. However, using a single routing configuration for long-time periods can be highly inefficient. Even more, the presence of abnormal and malicious traffic has magnified the network operation problem, claiming for solutions which not only deal with traffic uncertainty but also allow to detect and identify faulty traffic to take the appropriate countermeasures. In this paper, we introduce the Reactive Robust Routing (RRR) for TE, an approach that combines both proactive and reactive techniques to tackle the problem. Based on expected traffic patterns, we adapt the uncertainty set and build a multi-hour yet robust routing scheme that outperforms the stable robust approach. For the case of anomalous and unexpected traffic, we propose a fast anomaly detection/isolation algorithm to detect and localize abrupt changes in traffic flows and decide routing changes. This algorithm is optimal in the sense that it minimizes the decision delay for a given mean false alarm rate and false isolation probability. We validate these proposals using real data from two different backbone networks and we show how the RRR can handle uncertain and highly dynamic traffic in an automatic fashion, simplifying network operation.

**Index Terms**—Traffic Uncertainty, Multi-Hour Robust Routing, Anomaly Detection/Isolation, Reactive Robust Routing.

## I. INTRODUCTION

Traffic engineering (TE) represents a major issue for network operators in today’s scenario. TE allows the optimization of network resources usage through multiple mechanisms. In this work, we focus on routing optimization over an Autonomous System (AS). This optimization is becoming increasingly difficult due to the dynamic nature of current traffic. Traffic demands present two different components or *behaviors*: on one hand, a stable and predictable component due to usual traffic usage patterns (e.g. daily demand fluctuation); on the other hand, an abrupt and unpredictable behavior due to unexpected events, such as network equipment failures, flash crowds occurrences, security threats (e.g. denial of service attacks, virus propagation), external routing changes (e.g. inter-AS routing through BGP) and new spontaneous overlay services (e.g. P2P applications). We use the term *volume anomaly* [17] to describe these unexpected network events (large and sudden link load changes). Recent works [2]–[5]

have proposed a new perspective to the routing optimization under traffic uncertainty: the **Robust Routing (RR)** approach. In a robust fashion of TE, demand uncertainty is taken into account directly within the routing optimization, computing a single routing configuration for all demands within an *uncertainty set*. While this routing configuration is not optimal for any single traffic matrix (TM) within the set, it minimizes the worst case performance over the whole set. In this sense, RR provides performance guarantees (i.e. worst-case bounds) for all possible traffic variations within the uncertainty set. The RR approach can be used as a **proactive** technique to deal with dynamic traffic. It can handle changing demands at a reasonable cost (with respect to an ideal but illusive optimal adaptive routing) up to a certain limit (given by the size of the uncertainty set). However, applying a RR algorithm to address both traffic behaviors (usual traffic as well as volume anomalies) is an inefficient strategy: a single routing can not be suitable for both situations.

On the contrary, a **reactive** approach could be used as a complementary strategy to enhance RR performance, responding to abrupt and large traffic changes with an effective routing reconfiguration. Volume anomalies may have an important impact on the network performance, causing sudden situations of strong network congestion. The early detection and isolation of these anomalies allows to modify the routing as soon as possible, limiting their impact. In this work, we propose a signal processing algorithm for fast load change detection/isolation. Through out the paper, we use the term anomaly *isolation* to refer to the identification and localization of an anomalous flow among the network traffic.

### A. Related Work

There is a large literature on traffic engineering with uncertain traffic demands. Traditional algorithms rely on a small group of expected TMs (representative traffic demands from past observations) or estimated TMs to compute optimal and reliable routing configurations. An extreme case is presented in [11], where routing is optimized for a single estimated TM and it is then applied for long-time periods (24hs periods). Traffic uncertainty is characterized by multiple TMs in [12], [13] (e.g. set of TMs from previous day, same day of previous week, etc.), and different ways to find optimal routes for the set are presented. Given the dynamic nature of present demands,

this perspective is no longer suitable for current scenario [1]. A different approach is provided by online reactive algorithms: TeXCP [14] and MATE [15] both balance load in realtime, responding to instantaneous traffic demands. Their main goal is to avoid network congestion by adaptively balancing the load among paths, based on measurement. Reactive routing presents a desirable property, that of keeping routing adapted to current traffic. However, these adaptive algorithms present poor performance under significant and abrupt traffic changes [5]. A third category of algorithms consists in Stable Robust Routing techniques [2]–[6]. In [2], the authors capture traffic variations by introducing a polyhedral set of demands, applying linear programming techniques to compute an optimal stable routing for all demands within this set. [4] applies this robust technique to compute a robust MPLS routing configuration without depending on TM estimation, and discusses corresponding methods for robust OSPF optimization. Oblivious Routing [3] also defines linear algorithms to optimize worst-case performance for different sizes of traffic uncertainty sets, aiming to handle dynamic changes. [6] analyses the use of robust routing through a combination of traffic matrix estimation and its corresponding estimation error bounds, in order to shrink the uncertainty set. The drawback of stable robust routing is its inherent dependence on the definition of the uncertainty set: larger sets allow to handle a broader group of traffic demands, but at the cost of routing inefficiency; conversely, tighter sets produce more efficient routing schemes, but subject to poor performance guarantees. In [5], the authors introduce COPE, an approach to deal with this tradeoff in the size of the uncertainty set, combining traditional algorithms with oblivious routing. COPE optimizes routing for predicted demands and bounds worst-case performance to ensure acceptable efficiency under unexpected traffic events. Nevertheless, COPE proposes a long-term stable routing configuration as previous works do (24hs periods), losing the adaptability (and hence the performance efficiency) of reactive routing. Besides, it is possible not only to assure performance guarantees for unexpected events, but to obtain optimal routings for this traffic.

As regards anomaly detection in data networks, the problem has been extensively studied. In this section, we will just overview those works that have motivated our signal-processing based detection algorithm. Signal processing techniques have been applied to the anomaly detection field [7]–[9]. The usual behavior of data flows is modeled by several approaches: spectral analysis, time series analysis, wavelets decomposition, etc. Anomalies correspond to deviations from the usual behavior of the data flows. The general flaw of these algorithms is the lack of stability over time of the proposed traffic models, as well as the absence of optimality conditions for the detection in most cases. A second class of methods related to our model concerns statistical hypotheses testing [17]–[19]. When data flows are parametrically modeled, the design of optimal algorithms is possible. Nevertheless, non-parametric approaches are particularly studied because of the lack of parametric models, and these approaches are often sub-

optimal. The detection/isolation of traffic anomalies problem was previously treated in [17], using a TM decomposition on the Principal Component Analysis (PCA) basis. However, this approach presents a major stability problem: the PCA basis depends on the measurement period, rendering it unstable over time.

### B. Contributions of the Paper

In the final remarks of [4], the authors raised an interesting reflection: "it is not clear whether time-varying demands should be addressed using proactive (e.g. robust routing) or reactive (dynamic, adaptive) methods". In this work, we propose to use both proactive and reactive complementary approaches to deal with current dynamic traffic demands, separately treating both traffic uncertainty sources. For *expected traffic fluctuations*, we present a time varying approach of RR that outperforms the current *stable* approach: the **Multi-Hour Robust Routing (MHRR)**. The stable RR may be costly. However, it is easy to control its cost by shrinking the uncertainty set. We preserve the virtues of RR, but change the routing configuration during time. The uncertainty set is optimally divided into several uncertainty sub-sets that better adapt to real traffic loads, and a stable robust routing scheme is computed for each sub-set. The partitioning algorithm allows to optimally calculate the exact times when routing changes must be performed. For the case of *unpredictable traffic behavior*, we propose a novel volume anomaly detection/isolation algorithm to identify traffic problems and decide routing changes. This algorithm allows both to detect the volume anomaly and to identify and localize the anomalous flow. To overcome the stability problems of previous approaches, we propose a non data-driven traffic model which remains stable over time. The main contribution of this detection algorithm relies on the well established conditions of optimality that it presents, a feature generally absent in previous works.

Both proactive and reactive methods are combined into a novel TE approach for dynamic traffic demands: the **Reactive Robust Routing (RRR)**. This approach uses the MHRR to handle typical changes in traffic demands and the detection/isolation algorithm to deal with unexpected volume anomalies. The RRR exploits the isolation ability of the detection/isolation algorithm to compute an adapted robust routing configuration after the anomalous traffic detection, reducing its impact on network performance during its prevalence. In addition, it also provides a simple yet effective method to automatically detect the end of the anomaly, returning to the MHRR configuration. Contrary to previous works in the field, our proposal optimizes routing in a robust and adaptive fashion for every possible traffic demand (and not only for the common-case traffic). A key feature of the RRR approach relies on the fact that the whole routing configuration/reconfiguration algorithm is completely automatic, an interesting property that simplifies network operation by self-managing. All the proposed algorithms in this work are validated using real traffic data from two backbone networks, the Internet2 Abilene backbone network and a private international Tier-2 network.

The remainder of this paper is organized as follows. In Section II, we recall the basic aspects of the robust routing approach. Section III presents the theoretical background and empirical evaluation of the MHRR. The proposed algorithm and traffic model for anomaly detection/isolation are introduced and validated in section IV. Section V presents the Reactive Robust Routing, showing the automatic interaction between the proactive and the reactive components through complete real and simulated examples. Finally, Section VI concludes this work.

## II. ROBUST ROUTING

Let us consider a network topology defined by a set of  $n$  nodes and  $L = \{1, \dots, r\}$  links with capacities in  $C = (c_1, c_2, \dots, c_r)$ . The TM demand  $\mathbf{d} = \{d_{i,j}\}$  denotes the traffic flow between every node  $i$  and node  $j$  ( $i \neq j$ ) of the network. We re-arrange  $\mathbf{d}$  as a column vector,  $\mathbf{d} = \{d_k, k=1..m\}$ , where  $d_k$  represents the traffic flow transmitted by OD pair  $k$  (OD-flow  $k$ ) and  $m = n \times (n - 1)$  is the number of OD pairs. Let  $N = \{\text{OD}_1, \dots, \text{OD}_m\}$  be the set of OD pairs. Link's information  $y_l$  represents the total traffic (i.e. aggregated OD flows) through link  $l$  in a certain period of time. This information is available from router's MIB variables and it is usually collected every 5' periods via SNMP [20]. Traffic demands and links' traffic are related through the routing matrix  $R$ , a  $r \times m$  matrix which element  $0 \leq r_{l,k} \leq 1$  represents the fraction of OD demand  $k$  routed through link  $l$ :

$$\mathbf{y} = R \times \mathbf{d}. \quad (1)$$

with  $\mathbf{y} = \{y_l, l=1..r\}$ . Routing optimization depends on the underlying data transport mechanism; we will focus on path-based routing such as MPLS. This optimization consists in minimizing certain performance metric associated with traffic demand. Throughout this work we consider maximum link utilization (MLU) as the routing performance criterion. Overloaded links tend to cause QoS degradation (e.g. larger delays and packet losses, throughput reduction, etc.), so MLU represents a reasonable measure of network performance. For a given routing matrix  $R = \{r_{l,k}\}$  and a traffic demand  $\mathbf{d}$ , the MLU ( $u_{max}$ ) is defined as the maximum of the ratio between link load and link capacity:

$$u_{max}(C, \mathbf{d}, R) = \max_{l \in \{1..r\}} \sum_k \frac{r_{l,k} \cdot d_k}{c_l} = \max_{l \in \{1..r\}} \frac{y_l}{c_l} \quad (2)$$

Let  $P(k)$  be the set of possible paths for OD demand  $k$ . Let  $x_p^k$  be the proportion of traffic demand  $d_k$  that flows through path  $p \in P(k)$ ,  $0 \leq x_p^k \leq 1$ . Finally, let  $x_l^k$  be the proportion of traffic demand  $d_k$  that flows through link  $l \in L$ ,  $0 \leq x_l^k \leq 1$ . We define  $D$  as the uncertainty set where traffic demand may vary. This set can be defined in different ways, depending on the available information: link load measurements and historical routing, a set of previously observed TMs  $\{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^o\}$ , TM time series  $\mathbf{d}(t)$ , etc. [2] defines this set as a *polytope*, based on the intersection of several half-spaces that result from linear constraints imposed to traffic demand. The Robust Routing Optimization Problem (RROP) consists in minimizing  $u_{max}$ , considering all demands

within  $D$  (3). The RROP can be efficiently solved by linear programming techniques, applying a combined column and constraint generation method [2]. In a traditional robust routing

$$\begin{array}{ll} \text{minimize} & u_{max} \\ \text{subject to:} & \\ \sum_{p \in P(k)} x_p^k & \geq 1 \quad \forall k \in N \\ \sum_{p \in P(k), l \in p} x_p^k & \leq x_l^k \quad \forall k \in N, \forall l \in L \\ \sum_{k \in N} x_l^k \cdot d_k & \leq u_{max} \cdot c_l \quad \forall l \in L, \forall \mathbf{d} \in D \\ x_p^k, x_l^k & \geq 0 \quad \forall l \in L, \forall p \in P(k), \forall k \in N \\ u_{max} & \leq 1 \end{array} \quad (3)$$

application, the obtained routing configuration is applied during long-term periods of time (i.e. daily routing). In this sense, we refer to robust routing as **Stable Robust Routing (SRR)**.

## III. MULTI-HOUR ROBUST ROUTING

In [1] we present the advantages of the SRR with respect to traditional routing approaches: SRR offers stability guarantees against traffic uncertainty and traffic time-variations at a reasonable cost. However, considering a single routing scheme for long-time periods is conservative and results in sub-optimal performance. We propose a simple approach to shrink and adapt the uncertainty set along time that outperforms the SRR. Based on rough knowledge of traffic variations (i.e. considering expected traffic behavior), we propose to optimally divide the uncertainty set and build a multi-hour routing configuration, considering a single SRR configuration for each sub-set. Daily traffic changes can be seen as a time variation of

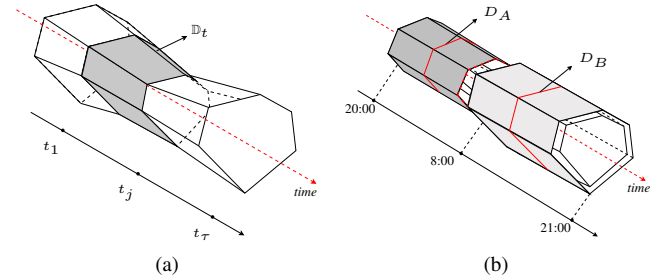


Fig. 1. (a) Daily variation of the polytope  $D_t$ . (b) time partitioning of  $D_t$ .

the uncertainty set. At each time  $t$ , the routing matrix  $R$  and the link load values  $\mathbf{y}(t) = \mathbf{y}^t$  define an instantaneous uncertainty set  $D(t) = \{\mathbf{d} \in \mathbb{R}^m, R \times \mathbf{d} \leq \mathbf{y}^t, \mathbf{d} \geq 0\}$ . The continuous union of infinite instantaneous uncertainty sets along time  $t$  defines the *daily uncertainty set*  $\mathbb{D}_t = \{(\mathbf{d}, t) \in \mathbb{R}^{m+1}, \mathbf{d} \in \cup_{t_1 \leq t \leq t_\tau} D(t), t_1 \leq t \leq t_\tau\}$ . Figure 1(a) explains this idea. Assuming this set is an union of polytopes, [10] provides a theoretical study of the optimal partitioning of  $\mathbb{D}_t$ , using a partitioning hyper plane. [10] proves that this is a NP-hard problem, except for the case where a partitioning direction is previously fixed. We define a partitioning hyper plane by its direction vector  $\alpha$  and a value  $w$ :  $\alpha \cdot \mathbf{d} = w$ . In the MHRR approach,

we consider a particular direction for partitioning: the *time direction*. In that case,  $w$  represents the time of the day. We define  $h + 1$  hyperplanes at times  $\{w_1, w_2, \dots, w_{h+1}\}$ . The intersection between  $\mathbb{D}_t$  and the half-spaces defined by these partitioning hyperplanes results in  $h$  uncertainty sub-sets  $\mathbb{D}_i = \{\mathbb{D}_t \cap \{\mathbf{d}, \alpha \cdot \mathbf{d} \geq w_i\} \cap \{\mathbf{d}, \alpha \cdot \mathbf{d} \leq w_{i+1}\}\}, \forall i = 1, \dots, h$ . Let  $D_i$  be the smallest single-time set that contains all demands  $\mathbf{d}(t) \in \mathbb{D}_i, w_i \leq t \leq w_{i+1}$  (see figure 1(b)). A SRR configuration  $R_{robust}^i$  is computed for each sub-set  $D_i$ . Each routing configuration is finally applied at each time interval. The optimal values of routing changes  $\mathbf{w}^* = \{w_2^*, \dots, w_h^*\}$  are the solution for the following optimization problem ( $w_1$  and  $w_{h+1}$  are fixed a priori, as they define the considered time interval of analysis):

$$\mathbf{w}^*(\mathbb{D}_t) = \arg \min_{\mathbf{w}} \left\{ \max_{i=1..h} u_{max}(D_i) \right\} \quad (4)$$

where  $u_{max}(D_i)$  is the solution for (3) for polytope  $D_i$ . [10] presents a simple algorithm to approximately solve (4) (within an arbitrary precision), using a generalization of a simple dichotomy methodology. The MHRR presents a trade-off between performance and routing stability. The more intervals we use, the more adapted the routing becomes. However, the number of intervals should be bounded as many routing changes may lead to instabilities and performance degradation. In a general case, 2 sub-sets are enough to handle the usual daily variation.

### MHRR Evaluation

We present a comparative analysis between SRR and MHRR in Abilene, an Internet2 backbone network. Abilene consists in 12 router-level nodes and 30 OC192 links (2 OC48). The used router-level network topology and traffic demands are available at [26]. Traffic data consists in 6-month traffic matrices collected every 5' via Netflow from the Abilene Observatory [27]. The time-variation of the polytope is not a simple homothety [1]; in this sense, we will show that a routing configuration change during the day improves routing performance. Let  $R_o$  be the historical routing matrix of Abilene, not necessarily optimal ( $R_o$  is available at [26]). We consider a single time partitioning (i.e. 2 routing intervals),  $w_1 = 20:00$ ,  $w_2 = w^*$  and  $w_3 = 21:00$ , where  $w^*$  is the solution for (4). For each time interval, we consider the smallest polytope that includes all possible realizations over that period:

$$D_{A,B} = \{\mathbf{d} \in \mathbb{R}^m, R_o \times \mathbf{d} \leq \mathbf{y}_{A,B}, \mathbf{d} \geq 0\} \quad (5)$$

where  $\mathbf{y}_A = \mathbf{y}_{max}^{20:00-w^*}$  and  $\mathbf{y}_B = \mathbf{y}_{max}^{w^*-21:00}$  (maximum values for each link). In this way,  $D_A$  includes all traffic demands between 20:00 and  $w^*$  and  $D_B$  between  $w^*$  and 21:00 (see figure 1(b)). For each polytope, we compute a SRR configuration,  $R_{robust}^A$  and  $R_{robust}^B$ . In order to compare stable and multi-hour approaches, we apply both routing configurations during the whole evaluation period. We include the routing performance obtained with  $R_o$  (curve *historical routing*) to appreciate the time variation of traffic loads. Figure 2(a) compares the routing performance (MLU) between these

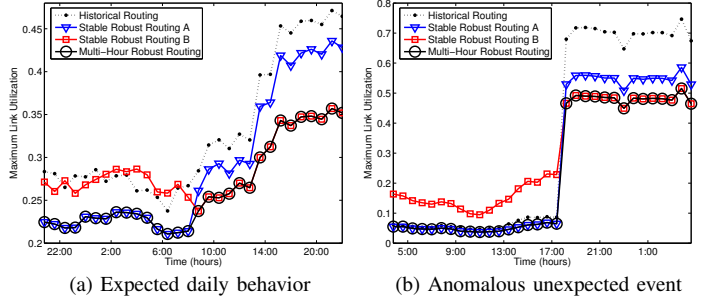


Fig. 2. Routing performance, stable vs. multi-hour robust routing.

two RR configurations. Polytope  $D_A$  is well suited for smaller loads, so  $R_{robust}^A$  performs better during the first half of the day, when network loading is lower. However, when traffic increases, demands that do not belong to  $D_A$  produce higher link utilizations than those obtained with  $R_{robust}^B$ . The MHRR consists in computing the time when routing must be changed ( $w^* \approx 8:00$  in this case), using the corresponding routing configuration depending on the time of the day ( $R_{robust}^A$  before  $w^*$  and  $R_{robust}^B$  after). The MHRR approach presents a performance improvement of 15% with respect to the SRR approach before  $w^*$ , reaching a near 20% of over-efficiency after  $w^*$ . We repeat the same evaluation but considering a traffic demand that drastically changes (i.e. a large time-variation of the polytope, caused by a volume anomaly). Figure 2(b) presents an abrupt change in MLU (almost 14 times higher) at time 18:00. In this case, we assume that this change is known in advance (note that in the general case, it is not possible to predict these abrupt changes). The optimal moment for changing routing is  $w^* \approx 18:00$ . The MHRR approach definitely outperforms the SRR in this experience, presenting a MLU between 10% and 60% smaller during the whole evaluation period.

## IV. DEALING WITH UNEXPECTED EVENTS

The proposed MHRR approach offers a robust and efficient routing configuration, provided a rough knowledge of the daily uncertainty set. However, in the presence of volume anomalies it is no longer possible to apply the MHRR as the daily uncertainty set is unknown. For those cases, we propose a fast volume anomaly detection/isolation algorithm to quickly identify faulty traffic. This detection allows to decide as soon as possible the moment when routing configuration must be changed. The goal of the algorithm is to detect/isolate an additive change  $\theta$  in the time series of traffic demand  $\mathbf{d}(t)$  from a sequence of link load measurements  $\mathbf{y}(t) = R \times \mathbf{d}(t)$ . We use link loads as input to avoid relying on seldom available traffic demands. In this work, we focus on detecting and isolating a “localized” anomaly<sup>1</sup>,  $\theta = \theta(\delta_{1,i}, \dots, \delta_{i,i}, \dots, \delta_{m,i})^T$ , where  $\delta_{i,j} = 0$  if  $i \neq j$  and  $\delta_{i,i} = 1$  (this corresponds to a change  $\theta$  in OD flow  $i$ ). The isolation of the anomalous traffic is possible

<sup>1</sup>If several OD flows are simultaneously corrupted, the detection/isolation algorithm produces an alarm and identifies only one faulty OD flow. The algorithm can be extended to detect simultaneous anomalies, but the complexity ( $n^0$  operations) grows like  $m^4$ , where  $m$  is the  $n^0$  of OD flows.



since an anomaly in a given OD flow typically spans multiple links. Real traffic demands follow a *non-observable* model from link load measurements: since  $r < m$ , it is impossible to retrieve  $\mathbf{d}(t)$  from  $\mathbf{y}(t)$  without additional assumptions on the traffic demand. To overcome this difficulty, we propose a parsimonious linear model for non-anomalous traffic. This model renders traffic demands observable and therefore, it allows to separate usual from anomalous traffic.

#### A. Stochastic Traffic Model for Anomaly Detection

We assume that the stochastic process of the OD traffic demand  $\mathbf{d}(t)$  obeys the following linear model:

$$\mathbf{d}(t) = \boldsymbol{\lambda}(t) + \boldsymbol{\xi}(t) \quad (6)$$

where  $\boldsymbol{\lambda}(t) \in \mathbb{R}^m$  is the mean traffic demand and  $\boldsymbol{\xi}(t)$  is a white Gaussian noise with covariance matrix  $\Sigma(t)$  that represents the model error. The process  $\boldsymbol{\lambda}(t)$  represents the “regular” part of the OD TM which can be correctly modeled when the behaviour of the network is anomaly-free. We propose to parameterize this vector by exploiting the stationarity of the spatial distribution of the TM. One of the few invariants of Internet traffic is that a small percentage of flows contribute to a large proportion of total traffic [4], [16]. Hence, if we assume that the traffic distribution between the different OD couples is spatially stationary in the absence of an anomaly, the order of increasing OD flows remains constant during long time periods. The proposed traffic model takes advantage of the stationary property of this ordering. We propose to classify OD flows in three different classes, depending on their volume: large OD flows, small OD flows and medium-size OD flows. The sorted components can be interpreted as a discrete increasing signal. The curve obtained by interpolating this discrete signal is assumed to be a continuous curve, hence it can be parameterized by using a polynomial approximation.

Figure 3 shows the OD flows, sorted in the increasing

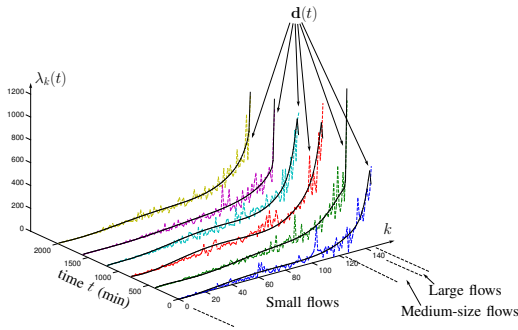


Fig. 3. Approximation of real OD flows by the spline-based model.

order of their volume of traffic, as a function of the time  $t$ . Since data are vectors of finite dimension, we propose to use the following method to design a discrete spline basis: (i) we choose a continuous spline basis; (ii) we discretize all these splines according to  $m$  points uniformly chosen on the interval  $[1; m]$  and (iii) we rearrange all these discrete signals according to previous sorting order. We finally obtain the

following linear model for the anomaly-free traffic demand:

$$\mathbf{d}(t) = S\boldsymbol{\mu}(t) + \boldsymbol{\xi}(t) \quad (7)$$

where  $S = (\mathbf{s}_1 \mathbf{s}_2 \dots \mathbf{s}_q)$  is a  $m \times q$  known matrix with columns  $\mathbf{s}_j$  and  $q$  is small with respect to  $m$ . The vectors  $\mathbf{s}_i$ , which correspond to the rearranged discrete spline, form a set of known basis vectors describing the spatial distribution of the traffic and  $\boldsymbol{\mu}(t) = (\mu_1(t) \dots \mu_q(t))^T$  is the unknown time varying parameter vector which describes the OD flow intensity distribution with respect to the set of vectors  $\mathbf{s}_i$ . The model for the anomaly-free link traffic is given by:

$$\mathbf{y}(t) = H\boldsymbol{\mu}(t) + \boldsymbol{\zeta}(t), \quad (8)$$

where  $H = RS$  and  $\boldsymbol{\zeta}(t) = R\boldsymbol{\xi}(t)$ . In this way, we can describe the usual behavior of traffic demands from simple link measurements. The computation of the rank of  $H$  is not simple since it depends on the routing matrix  $R$ . In practice, since the number of columns of  $H$  is very small, the product  $RS$  and its rank can be computed very fast. Therefore, we will assume that  $H$  is full column rank. Finally, the covariance matrix  $\Sigma$  is unknown. The remedy consists in computing an estimate  $\hat{\Sigma}$  of  $\Sigma$ . Results on the estimation of  $\hat{\Sigma}$  can be found in [22].

#### B. Volume Anomaly Detection/Isolation

The detection/isolation of a volume anomaly at time  $t_0$  can be treated as a hypothesis testing problem where the null hypothesis  $\mathcal{H}_{t_0}^0 = \{\text{the OD flows are anomaly-free at time } t_0\}$  is tested against  $m$  alternatives  $\mathcal{H}_{t_0}^j = \{\text{the } j\text{-th OD flow presents an anomalous additional amount of traffic } \theta \text{ from the time } t_0\}$ . The change detection algorithm has to compute a pair  $(T, \nu)$ , where  $T$  is the alarm time at which a  $\nu$ -type change ( $\nu \in \{1, 2, \dots, m\}$ ) is detected and isolated, based on link traffic observations  $\mathbf{y}_1, \mathbf{y}_2, \dots$ . This algorithm is optimal in the sense that it minimizes the maximum mean delay for detection/isolation, for a given minimum mean time before a false alarm  $\gamma$  and maximum false isolation probability  $\beta$ , both defined by an expert user. The hypothesis testing can be written as

$$\begin{aligned} \mathcal{H}_0 &: \mathbf{y}(t) \sim \mathcal{N}(H\boldsymbol{\mu}(t), R\Sigma R^T), \quad t = 1, 2, \dots, \\ \mathcal{H}_{t_0}^j &: \begin{cases} \mathbf{y}(t) \sim \mathcal{N}(H\boldsymbol{\mu}(t), R\Sigma R^T), \quad t = 1, \dots, t_0 - 1, \\ \mathbf{y}(t) \sim \mathcal{N}(H\boldsymbol{\mu}(t) + \theta_j \mathbf{r}_j, R\Sigma R^T), \\ \theta_{j,1} \leq |\theta_j| \leq \theta_{j,2}, \quad t = t_0, t_0 + 1, \dots \end{cases} \end{aligned} \quad (10)$$

where  $\mathbf{r}_j, 1 \leq j \leq m$  denotes the normalized  $j$ -th column of  $R$  and  $0 < \theta_{j,1} < \theta_{j,2} < +\infty$  are some known bounds on the change intensity of the  $j$ -th OD flow (these bounds are introduced for technical reasons but they can be chosen arbitrarily). As we show in the Appendix, we can simplify this problem by eliminating the non-anomalous traffic. In this case, hypothesis (10) can be rewritten as

$$\mathcal{H}_{t_0}^j : \begin{cases} \mathbf{z}(t) \sim \mathcal{N}(0, I_{r-q}), \quad t = 1, \dots, t_0 - 1, \\ \mathbf{z}(t) \sim \mathcal{N}(\theta_j \mathbf{v}_j, I_{r-q}), \\ \theta_{j,0} \leq |\theta_j| \leq \theta_{j,1}, \quad t = t_0, t_0 + 1, \dots \end{cases} \quad (11)$$

where  $\mathbf{v}_j$  is a known vector and  $\mathbf{z}(t)$  are the normalized residuals obtained from  $\mathbf{y}(t)$  after filtering the non-anomalous traffic. The vector  $\mathbf{v}_j$  corresponds to the signature in the

residuals of a change in OD flow  $j$ . We use the optimal recursive algorithm  $(T_r, \nu_r)$  proposed in [25] to solve (11) :

$$T_r = \min_{1 \leq k \leq m} \{T_r(k)\}, \quad \nu_r = \arg \min_{1 \leq k \leq m} \{T_r(k)\}$$

$$T_r(k) = \inf \left\{ t \geq 1 : \min_{0 \leq j \neq k \leq m} [g_t(k, j) - h_{k,j}] \geq 0 \right\} \quad (12)$$

with  $g_t(k, j) = g_t(k, 0) - g_t(j, 0)$ . The recursive functions  $g_t(k, 0)$  are defined by

$$g_t(k, 0) = (g_{t-1}(k, 0) + z_t(k, 0))^+ \quad (13)$$

$$z_t(k, 0) = \log \frac{f_k(\mathbf{z}(t))}{f_0(\mathbf{z}(t))} \quad (14)$$

$g_0(k, 0) = 0$  for every  $1 \leq k \leq m$  and  $g_t(0, 0) = 0$  for all  $t$ .  $f_0$  represents the probability density function of anomaly-free traffic measurements.  $f_k$  is the probability density function of residuals  $\mathbf{z}(t_0), \mathbf{z}(t_0 + 1), \dots$  after a change of type  $k$ . The thresholds  $h_{k,j}$  are chosen by the following formula:

$$h_{k,j} = \begin{cases} h_d & \text{if } 1 \leq k \leq m \quad \text{and } j = 0 \\ h_i & \text{if } 1 \leq k, j \leq m \quad \text{and } j \neq k \end{cases}$$

where  $h_d$  is the detection threshold and  $h_i$  is the isolation threshold. For given bounds  $\gamma$  and  $\beta$ , this algorithm is asymptotically optimal, i.e. it reaches the lower bound of the maximum mean delay for detection [25]. The choice of the detection and isolation thresholds  $h_d$  and  $h_i$  is discussed (with practical comments and simulations) in [24].

### C. Validation

We demonstrate the ability of the detection/isolation algorithm to detect and identify a volume anomaly in SNMP link flow data from two different networks (different not only in the topology but also in the behavior of traffic demands): a large Tier-2 network (50 nodes, 168 measured links and 2450 non-zero OD flows, sampled at a 10 minute rate) and Abilene (the Abilene dataset consists in Netflow traces, so we use the supplied routing matrix to retrieve link loads). Figure 4 shows the typical realizations of the decision functions  $g_t(i, 0)$  and  $s_t(i) = \min_{0 \leq i \neq k \leq m} [g_t(i, k) - h_{i,k}]$  vs the elapsed time. The functions  $s_t(i)$  are used to “monitor” the OD flows; when the function  $s_t(i)$  exceeds 0, OD flow  $i$  is declared faulty. It is assumed that the anomaly in the Tier-2 network begins at time 3660, and at time 1070 in Abilene. Note that after this time, several decision functions  $g_t(i, 0)$  rapidly grow. Each function  $g_t(i, 0)$  is associated with OD flow  $i$  and when this function grows, it means that OD flow  $i$  is suspected of carrying an abnormal amount of traffic. Contrary to  $g_t(i, 0)$ , only decision function  $s_t(159)$  ( $s_t(87)$  in Abilene) associated to faulty OD flow 159 (87 respectively) grows and finally exceeds the threshold. Hence, the functions  $s_t(i)$  permit us to isolate the faulty OD flow among all the OD flows associated to functions  $g_t(i, 0)$  that have rapidly grown. At time 3660 (1070 respectively), an alarm is raised and the algorithm selects the faulty OD flow 159 (87 respectively). The decision function  $s_t(i)$  needs only 1 observation (10 minutes in the Tier-2 network or 5 minutes in Abilene, but this is the smallest delay than can be achieved given these sampling-rates) to detect and isolate the faulty OD flow. An interesting observation of this

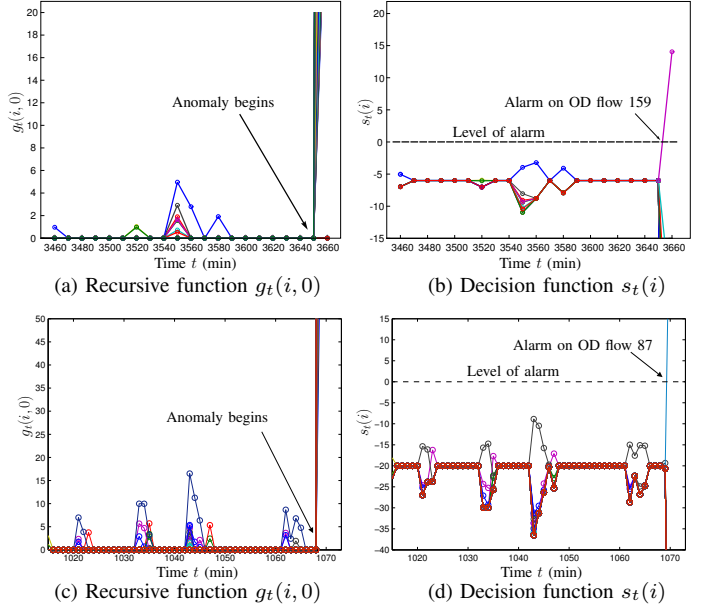


Fig. 4. Typical realizations of decision functions for a Tier-2 network (a,b) and Abilene (c,d).

evaluation is that the detection/isolation algorithm achieves good results in both networks, even though the respective traffic demand behaviors are completely different between these two networks.

## V. REACTIVE ROBUST ROUTING

Both proactive and reactive methods (the MHRR and the anomaly detection/isolation algorithm respectively) are combined into a single approach we refer to as the Reactive Robust Routing (RRR). This approach provides an **automatic** method for robust routing configuration/reconfiguration, based on the monitoring of the network state. The RRR exploits the isolation ability of the detection/isolation algorithm to compute a new robust routing configuration after the detection of an anomalous OD flow; at the same time, it detects the end of the anomaly (if there is any) and returns to the usual MHRR routing.

### A. Routing Reconfiguration

We propose a simple method that exploits both the RR approach and the isolation ability of previous detection/isolation algorithm to compute the new routing scheme to apply after the detection step. The idea of this reconfiguration is to minimize the impacts of the detected anomaly on the network performance. We assume that before the detection of the anomalous traffic, a stable RR configuration  $R_{robust}^A$  is applied, computed on the basis of the historical routing  $R_o$  and the link load  $\mathbf{y}_o$  that results from the MHRR algorithm (5) ( $R_{robust}^A$  is obtained from (3), using  $D_A = \{\mathbf{d} \in \mathbb{R}^p, R_o \times \mathbf{d} \leq \mathbf{y}_o, \mathbf{d} \geq 0\}$ ). After the detection and isolation of the faulty OD flow  $k$ , the anomalous-free traffic demand  $\mathbf{d}$  takes the value  $\mathbf{d}^* = \mathbf{d} + \boldsymbol{\theta}$ , with  $\boldsymbol{\theta} = \theta \cdot \boldsymbol{\delta}_k$ , where  $\boldsymbol{\delta}_k = (\delta_{1,k}, \dots, \delta_{k,k}, \dots, \delta_{p,k})^T$ ,  $\delta_{i,k} = 0$  if  $i \neq k$  and  $\delta_{k,k} = 1$ . We can expand the uncertainty set  $D_A$  in the directions of the routed OD

flow  $k$  (with respect to  $R_o$ , i.e. the routing configuration that defined  $D_A$ ), obtaining an **expanded uncertainty set**  $D_C = \{\mathbf{d}^* \in \mathbb{R}^P, R_o \times \mathbf{d}^* \leq \mathbf{y}_o + R_o\theta, \mathbf{d} \geq 0\}$ . The reader should bear in mind that the type of anomalies we deal with are generally caused by external factors (e.g. external routing changes, flash crowds, denial of service attacks); this justifies the relevance of the uncertainty set expansion with respect to  $R_o$ . Said in other words, we detect and identify which is the anomalous OD flow, and then we consider a bigger uncertainty set that takes into account the abrupt change of this OD flow. The new RR scheme  $R_{robust}^C$  is the solution for (3), using  $D_C$ . To avoid the estimation of the unknown anomalous volume  $\theta$ , we can expand  $D_A$  to the limits of links' capacities, in the direction of OD flow  $k$ :  $D_C = \{\mathbf{d}^* \in \mathbb{R}^P, R_o \times \mathbf{d}^* \leq \mathbf{y}_o \cdot \bar{\zeta}_{i,k} + C \cdot \zeta_{i,k}, \mathbf{d} \geq 0\}$ , where  $\zeta_{i,k} = 1$  if OD flow  $k$  is routed across link  $i = 1..r$  and  $\bar{\zeta}_{i,k} = 0$  otherwise, and  $\bar{\cdot}$  denotes the ones' complement of  $\cdot$  (bitwise NOT, i.e.  $0 \rightarrow 1$  and  $1 \rightarrow 0$ ). While the outcome of this approach may result in routing inefficiency, it avoids the estimation errors of  $\theta$  (i.e. we build a more robust routing).

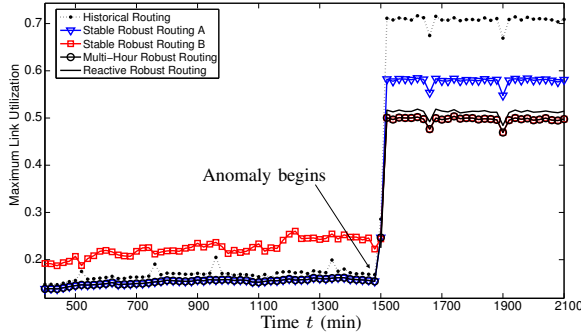


Fig. 5. Reactive Robust Routing - routing reconfiguration after the detection of a large and abrupt traffic change.

Figure 5 presents the evaluation of the RRR approach in the presence of a sudden and abrupt load change. We consider the same situation of figure 2(b), comparing the routing performance of the MHRR and the RRR respectively. As in section III, we assume the daily uncertainty set is completely known for the case of the MHRR (i.e. the abrupt change is known in advance). For the RRR, the anomaly is automatically detected and the new routing configuration is computed and immediately applied, based on the expansion of the uncertainty set. We can appreciate that the routing performance of the RRR is slightly worse than the one obtained with the MHRR (less than 2%). Nevertheless, the RRR represents a real scenario, where the anomaly can not be forecasted and has to be detected to compute an accurate rerouting.

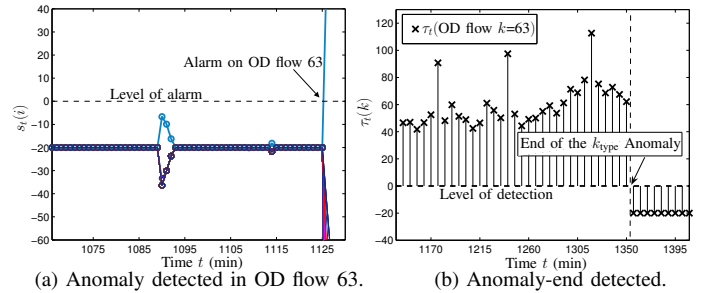
### B. Back to the MHRR scenario

After the anomaly detection and the robust routing re-configuration, we must provide a way to detect the end of the anomaly, in order to return to the MHRR situation. This detection can be easily achieved by using a simplified version of our detection algorithm: suppose that we detect and isolate

a  $k_{\text{type}}$  anomaly at time  $t_0$  (i.e. OD flow  $k$  is declared as anomalous). For every time  $t > t_0$ , we only monitor OD flow  $k$  until no anomaly alarms are raised, showing the end of this anomaly (remember that in this work we have only considered “localized” anomalies, i.e. anomalies in a single OD flow at a time). As we focus on a single OD flow, the multi-hypotheses test (9), (10) becomes a single hypothesis test, where the null hypothesis  $\mathcal{H}_t^0 = \{\text{the OD flow } k \text{ is anomaly-free at time } t\}$  is tested against  $\mathcal{H}_t^1 = \{\text{the } k\text{-th OD flow presents an anomalous additional amount of traffic at time } t\}$  (note that this hypothesis test is slightly modified, as both hypothesis are exchanged). Instead of using a sequential probability test (e.g. single CUSUM algorithm [23]), we propose to apply a simple LLR test (Log Likelihood Ratio Test) at each time  $t$ . For this purpose, we consider the decision function  $\tau_t(k)$

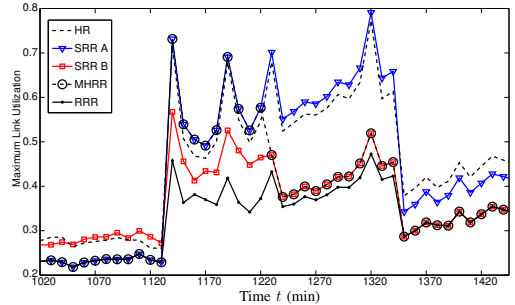
$$\tau_t(k) = \left( \log \frac{f_k(\mathbf{z}(t))}{f_0(\mathbf{z}(t))} \right)^+ - h_d \quad (15)$$

In the presence of an anomalous behavior in OD flow  $k$ ,  $\tau_t(k) > 0$  and the anomaly alarm keeps raising. When  $\tau_t(k)$  becomes negative, the anomaly alarm ceases, pointing out its end. To conclude this section, we present in figure



(a) Anomaly detected in OD flow 63.

(b) Anomaly-end detected.



(c) Reactive Robust Routing.

Fig. 6. Reactive Robust Routing performance under a simulated DoS attack.

6 an evaluation of the complete RRR approach under the presence of a volume attack (e.g. single DoS attack). We introduce an artificial sudden and large volume change in OD flow 63 of the Abilene dataset. This artificial traffic is put on top of the usual daily traffic between times 1125 and 1350. The first step of the RRR consists in computing the MHRR, using an expected daily uncertainty set. The optimal division (4) results in  $w^* = 1230$ . The evaluation begins at time 1020, when the MHRR decides to apply the SRR  $R_{robust}^A$  (SRR A in fig. 6). The detection/isolation algorithm continuously monitor the network state, and at time  $t_0 = 1125$



detects and localizes an anomalous behavior in OD flow 63 (figure 6(a)). After the detection (and before the new sampling of link loads  $\mathbf{y}(t_0 + 1)$ , i.e. a 5' time-window) the new routing configuration is computed, according to V-A. At time  $t = t_0 + 1$  the new routing configuration is deployed and the anomaly-end detection phase begins. It is important to note that the matrix  $H = RS$  as well as the *usual-traffic rejector* (see the Appendix) must be recomputed after the change of the routing matrix  $R$  (in fact, the same re-computation must be conducted every time the routing matrix changes, restarting the detection algorithm to avoid transient effects). The decision function  $\tau_t(63)$  remains positive for every time  $t > t_0$ , until time  $t' = 1350$ , when the negative value of  $\tau_{t'}(63)$  shows the end of the anomalous behavior in OD flow 63. At this time, the RRR compares  $t'$  with  $w^*$  in order to decide which routing to apply, whether SRR  $R_{robust}^A$  or  $R_{robust}^B$  ( $R_{robust}^A$  if  $t' < w^*$  or  $R_{robust}^B$  if  $t' > w^*$ ). Once the new routing configuration is established, the anomaly detection/isolation algorithm starts again to search for anomalous behaviors. The performance improvements of the RRR are evident, up to a 40% wrt the MHRR and near 50% wrt the traditional SRR approach.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we address the routing optimization under traffic uncertainty problem. We provide a solution that not only deals with current dynamic traffic demands in a robust and efficient way but also detects and isolates large-volume anomalous traffic, improving network operation. We extend the robust routing paradigm by introducing the notion of time-varying uncertainty set, setting up a multi-hour robust routing scheme. We show that this approach achieves better resource utilization than previous stable robust proposals in different scenarios. We introduce an original linear spline-based parsimonious model to parameterize usual traffic behavior from widely available link load measurements. Compared to many other traffic models, ours remains stable along time, a necessary condition to achieve reliable results. Based on this model, we present a statistical algorithm to detect and isolate volume anomalies in network traffic. This algorithm presents well-established conditions of optimality, unavailable in previous proposals in the field. We apply this algorithm to cope with sudden and large traffic changes in current dynamic demands, complementing the multi-hour robust scheme. We propose a simple method that exploits both the RR approach and the isolation ability of previous detection/isolation algorithm to compute the new routing scheme to apply after the detection step. The idea of this reconfiguration is to minimize the impacts of the detected anomaly on the network performance. All these algorithms are merged into a new proposal for robust and reactive routing optimization, the Reactive Robust Routing. The RRR approach deals with traffic uncertainty in a completely automatic fashion, simplifying network management.

We believe that the RRR represents a first step towards a dynamic and robust routing policy, but many important issues remain open for further study. A deep evaluation of

the impact of routing re-configuration on end-to-end traffic must be conducted, especially considering the imposed QoS restrictions in the actual end-user Internet-services scenario.

## REFERENCES

- [1] P. Casas and S. Vaton, "An Adaptive Multi Temporal Approach for Robust Routing", in *Proc. Euro-FGI Workshop on IP QoS and Traffic Control*, 2007.
- [2] W. Ben-Ameur and H. Kerivin, "Routing of Uncertain Traffic Demands", in *Optimization and Engineering*, 2005.
- [3] D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs", in *Proc. ACM SIGCOMM*, 2003.
- [4] M. Johansson and A. Gunnar, "Data-driven Traffic Engineering: techniques, experiences and challenges", in *Proc. BROADNETS 2006*, 2006.
- [5] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang and A. Greenberg, "COPE: Traffic Engineering in Dynamic Networks", in *Proc. ACM SIGCOMM*, 2006.
- [6] I. Juva, "Robust Load Balancing", in *Proc. IEEE GLOBECOM*, 2007.
- [7] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies", in *ACM SIGCOMM Internet Measurement Workshop*, 2002.
- [8] Brutlag, Jake D., "Aberrant Behavior Detection in Time Series for Network Monitoring", in *Proc. 14th Systems Administration Conference*, 2000.
- [9] C.M. Cheng, H. Kung, K.S. Tan, "Use of Spectral Analysis in Defense Against DoS Attacks", in *Proc. IEEE GLOBECOM*, 2002.
- [10] W. Ben-Ameur, "Between Fully Dynamic Routing and Robust Stable Routing", in *Proc. DRCN 2007*, 2007.
- [11] M. Roughan, M. Thorup and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices", in *Proc. USENIX/ACM Internet Measurement Conference*, 2003.
- [12] C. Zhang, Z. Ge, J. Kurose, Y. Liu and D. Towsley, "Optimal Routing with Multiple Traffic Matrices: Tradeoff between Average case and Worst case Performance", in *Proc. 13th International Conference on Network Protocols (ICNP)*, 2005.
- [13] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll and D. Towsley, "On Optimal Routing with Multiple Traffic Matrices", in *Proc. IEEE INFOCOM*, 2005.
- [14] S. Kandula, D. Katabi, B. Davie and A. Charny, "Walking the Tightrope: Responsive yet Stable Traffic Engineering", in *Proc. ACM SIGCOMM*, 2005.
- [15] A. Elwalid, C. Jin, S. Low and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", in *Proc. IEEE INFOCOM*, 2001.
- [16] A. Medina, K. Salamatiyan, S. Bhattacharyya and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *Proc. ACM SIGCOMM*, 2002.
- [17] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies", in *Proc. SIGCOMM*, 2004.
- [18] A. Lakhina et al., "Characterization of Network-Wide anomalies in Traffic Flows", in *IMC*, 2004.
- [19] M. Thottan and C. Ji, "Anomaly Detection in IP Networks", in *IEEE Trans. on Signal Processing*, 2003.
- [20] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford and F. True, "Deriving Traffic Demands for Operational IP Networks", in *IEEE/ACM Trans. on Networking*, 2001.
- [21] C. Rao, "Linear Statistical Inference and its Applications", John Wiley & Sons.
- [22] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data", in *Journal of American Statistical Association*, 1996.
- [23] M. Basseville and I. Nikiforov, "Detection of abrupt changes: theory and applications", Prentice Hall, 1993.
- [24] I. Nikiforov, "A simple recursive algorithm for diagnosis of abrupt changes in random signals", in *IEEE Trans. on IT*, 2000.
- [25] I. Nikiforov, "A lower bound for the detection/isolation delay in a class of sequential tests", in *IEEE Trans. on IT*, 2003.
- [26] Y. Zhang, "Abilene Dataset 04", <http://www.cs.utexas.edu/yzhang/research/AbileneTM/>.
- [27] The Abilene Observatory, <http://abilene.internet2.edu/observatory/>.

## APPENDIX - ELIMINATION OF NON-ANOMALOUS TRAFFIC

Non-anomalous traffic  $H\boldsymbol{\mu}(t)$  is eliminated by projecting the measurement vector  $\mathbf{y}(t)$  on the null space of  $H$ . By using the invariant properties of the Gaussian law, the general covariance matrix in (10) is reduced to the identity one. Let us define the matrix  $W = (\mathbf{w}_1, \dots, \mathbf{w}_{r-q})$  of size  $r \times (r - q)$  composed of eigenvectors  $\mathbf{w}_1, \dots, \mathbf{w}_{r-q}$  of the projection matrix  $P_H^\perp = I_r - H(H^T H)^{-1} H^T$  corresponding to eigenvalue 1. The matrix  $W$  satisfies the following conditions:  $W^T H = 0$ ,  $W W^T = P_H^\perp$  and  $W^T W = I_{r-q}$ . The matrix  $W$  can be considered as a linear rejector that eliminates the non-anomalous traffic. Under hypothesis  $\mathcal{H}_{t_0}^0$ , the sequence  $W^T \mathbf{y}(t)$  can be modeled as  $W^T \mathbf{y}(t) = W^T \boldsymbol{\zeta}(t) + \theta_j W^T \mathbf{r}_j$ ,  $j = 1, \dots, m$ . Since  $W^T \boldsymbol{\zeta}(t)$  is a correlated Gaussian vector with covariance matrix  $\tilde{\Sigma} = W^T R \Sigma R^T W$ , each vector  $W^T \mathbf{y}(t)$  is normalized by using the square root matrix  $\tilde{\Sigma}^{\frac{1}{2}}$ ,  $\mathbf{z}(t) = \tilde{\Sigma}^{-\frac{1}{2}} W^T \mathbf{y}(t) \sim \mathcal{N}(\theta_j \mathbf{v}_j, I_{r-q})$ , with  $\mathbf{v}_j = \tilde{\Sigma}^{-\frac{1}{2}} W^T \mathbf{r}_j$ .