EuroFGI Workshop on IP QoS and Traffic Control
P. Pereira (Ed.)
Lisbon, Portugal, December 6–7, 2007

AN ADAPTIVE MULTI-TEMPORAL APPROACH FOR ROBUST ROUTING

Pedro Casas* and Sandrine Vaton*

* ENST Bretagne
e-mail: pedro.casas@enst-bretagne.fr, sandrine.vaton@enst-bretagne.fr

Keywords: Traffic Engineering, Traffic Matrix, Robust Routing (RR), Multi-Temporal RR.

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 deal with traffic uncertainty, computing a stable routing configuration that is immune to traffic variations within certain uncertainty set. While this robust approach achieves routing reliability at low optimality loss, using a single routing configuration for long-time periods can be inefficient. Based on expected traffic patterns, we show that it is possible to adapt the uncertainty set and build a multi-temporal yet robust routing scheme that outperforms the stable approach. This work presents the study of robust routing in a real network topology, exploring the tradeoffs between stable and multi-temporal robust routing.*

1 INTRODUCTION

Traffic Engineering (TE) represents a major issue for Network Operators in today's scenario. TE allows the optimization of network resources usage through different 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 variations present not only a slow predictable component due to normal traffic usage patterns (e.g. daily demand fluctuation) but also an abrupt and unpredictable behaviour 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). Traditionally considered approaches are based on traffic matrix (TM) estimation and prediction. We classify them as stable routing and *load-sensitive* routing (dynamic routing from now on). They both present some conception drawbacks that render them unsuitable for current scenario; the former proposes a single-time solution that relies on seldom available traffic knowledge, the latter is complex to implement and induces potential instabilities.

1.1 Stable routing optimization and traffic matrix estimation

Routing optimization depends on the underlying data transport mechanism; we will focus on path-based routing such as MPLS. Assuming a single known value of the origin/destination (OD) TM, the stable routing optimization consists in balancing the load over a certain number of OD paths in order to minimize/maximize some performance criterion. This is a well known multi-commodity flow problem, easily solved by linear programming techniques. However, even though current networks perform flow measurements, these are not always conducted on all links and egress/ingress points of the network and real TMs are normally unavailable. Moreover, in order to avoid CPU router exhaustion, router vendors have implemented sampled versions of flow-level measurement protocols, resulting in potentially large errors in volume estimation. Thus, routing optimization is often computed for an estimated TM, resulting in a sub-optimal routing configuration and, depending on the goodness of the estimation, highly inefficient for the real TM.

TM estimation consists in estimating the demand for each OD pair of the network from routing and links' information. Given a network topology defined by a set of n nodes and r links, the traffic matrix 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 and $m = n \times (n-1)$ is the number of OD pairs. Link's information y_l represents the total traffic 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 [7]. Traffic demands and links' traffic are related through the routing matrix R, a $r \times m$ matrix which element $0 \leqslant r_{l,k} \leqslant 1$ represents the fraction of OD demand k routed through link l:

$$\mathbf{y} = R \times \mathbf{d}. \tag{1}$$

with $\mathbf{y} = \{y_{l,\ l=1..r}\}$. R represents the routing configuration at the time of link load measurements (i.e. it is known). This system of linear equations is ill posed $(m >> r)$, thus the computation of demand $\mathbf{d}$ becomes in fact a problem of estimation. Different techniques are applied to solve this estimation problem, a brief survey is presented in [6].

1.2 Dynamic routing under highly dynamic traffic

Previous stable routing computes a single-time routing configuration. Dynamic TE consists in using this configuration during a certain period of time, re-computing a new configuration when traffic changes become *noticeable* (i.e. it adapts the routing configuration to traffic variations). Dynamic TE assumes that traffic is stable and so, routing reconfiguration does not occur very often. However, dynamic routing presents a great handicap: adapting to the highly variable and unpredictable current traffic has an undeniable associated cost, that of potential instabilities.

Recent works [1–4] have proposed a new perspective to the routing optimization under traffic uncertainty problem: the **Robust Routing (RR)** approach. In a robust approach of TE, demand uncertainty is taken into account directly into 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 TM within the set, it minimizes the worst case performance over the whole set. RR provides performance guarantees (i.e. worst-case bounds) for all possible traffic variations within the uncertainty set. However, applying a single robust configuration in the presence of highly variable traffic raises a difficult question: how should this uncertainty set be defined? Larger sets cover a broader group of possible demands, but at the cost of routing inefficiency. On the other hand, tighter sets produce more efficient routing schemes, but subject to poor performance guarantees.

1.3 Contributions of the paper

We present a novel time varying perspective for RR that outperforms the current *stable* approach: the **Multi-Temporal Robust Routing (MTRR)**. 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 RR scheme is computed for each sub-set. The partitioning algorithm allows to calculate the exact moments when routing changes must be performed, simplifying the network operation. This proposal is validated using real traffic data from the Internet2 Abilene backbone network [8].

The remainder of this paper is organized as follows. In Section 2, the robust routing approach is introduced, analyzing its main features through real network examples. Section 3 presents the theoretical background of the MTRR. An empirical evaluation of the MTRR in the Abilene backbone network is presented in Section 4. Finally, Section 5 concludes this work.

2 ROBUST ROUTING

Good TE policies must have the ability to deal with traffic uncertainty, providing reasonable and *ensured* performance levels even in the case of unexpected events. A major progress in this direction was achieved in [1] by applying robust optimization techniques to the routing under traffic uncertainty problem. The idea is to take into consideration all possible traffic matrices within some bounded set. [1] defines this set as a *polytope*, based on the intersection of several half-spaces that result from some linear constraints imposed to traffic demands. An interesting aspect of robust routing relies on the fact that, given a polytope, a single routing configuration is computed, avoiding possible instabilities due to routing changes. In this sense, we refer to robust routing as **Stable Robust Routing (SRR)**.

2.1 Problem formulation

Let us consider a network topology defined by a set of n nodes and r unidirectional links with capacities in $C = (c_1, c_2, \ldots, c_r)$. For the sake of clarity, we will first describe the traditional routing optimization problem, assuming a single, known traffic demand $\mathbf{d}$. This optimization consists in minimizing certain performance metric associated with demand $\mathbf{d}$. 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:

$$u_{max}\left(C, \mathbf{d}, R\right) = \max_{l\in\{1\ldots r\}} \sum_{k} \frac{r_{l,k}\cdot d_k}{c_l} = \max_{l\in\{1\ldots r\}} \frac{y_l}{c_l} \tag{2}$$

Let $N = \{1, \ldots, m\}$ be the set of OD pairs and $L = \{1, \ldots, r\}$ the set of links. 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 $p \in P(k)$, $0 \leqslant x_p^k \leqslant 1$. Finally, let x_l^k be the proportion of traffic demand d_k that flows through link $l \in L$, $0 \leqslant x_l^k \leqslant 1$. Table (1.a) presents the traditional multipath routing optimization problem:

minimize	u_{max}		(3)	minimize	u_{max}		(8)
subject to:				subject to:			
$\sum\limits_{p\in P(k)} x_p^k$	$\geqslant 1$	$\forall\, k \in N$	(4)	$\sum\limits_{p\in P(k)} x_p^k$	$\geqslant 1$	$\forall\, k \in N$	
$\sum\limits_{p\in P(k),\, l\in p} x_p^k$	$\leqslant x_l^k$	$\forall\, k \in N,\ \forall\, l \in L$	(5)	$\sum\limits_{p\in P(k),\, l\in p} x_p^k$	$\leqslant x_l^k$	$\forall\, k \in N,\ \forall\, l \in L$	
$\sum\limits_{k\in N} x_l^k.d_k$	$\leqslant u_{max}\cdot c_l$	$\forall\, l \in L$	(6)	$\sum\limits_{k\in N} x_l^k.d_k$	$\leqslant u_{max}\cdot c_l$	$\forall\, l \in L,\ \forall\, \mathbf{d} \in D$	(9)
x_p^k	$\geqslant 0$	$\forall\, p \in P(k),\ \forall\, k \in N$		x_p^k	$\geqslant 0$	$\forall\, p \in P(k),\ \forall\, k \in N$	
x_l^k	$\geqslant 0$	$\forall\, l \in L,\ \forall\, k \in N$		x_l^k	$\geqslant 0$	$\forall\, l \in L,\ \forall\, k \in N$	
u_{max}	$\leqslant 1$		(7)	u_{max}	$\leqslant 1$		
	(a)				(b)		

Table 1: (a) Single routing optimization, (b) RR optimization.

All problem constraints are linear. Constraints (4) express the multipath property of the routing (each OD flow can be transmitted through different paths, and every flow must be completely routed). Constraints (5) simply define the link traffic proportions x_l^k. While these additional variables are not necessary, they are introduced for an easier comprehension of the problem formulation. As a side effect, routing $\{r_{l,k}\} = \{x_l^k\}$ is directly obtained. Constraints (6) define the concept of u_{max}. Finally, constraint (7) specifies that routing must be stable. From an algorithmic point of view, this is an easy to solve linear programming problem. However, as we previously stated, real values of traffic demand are not always available, and all we can expect is to find this real value within some bounded uncertainty set.

Let us now consider the robust case, where traffic demand $\mathbf{d}$ belongs to certain polytope of traffic demands D. The Robust Routing Optimization Problem (RROP) consists in minimizing u_{max}, considering all demands within the polytope D. Table (1.b) presents the RROP formulation. At first sight, it looks like a really difficult problem. Indeed, traffic demand's uncertainty largely modifies the traditional problem; constraints (9) are no longer linear and the polytope D is generally an infinite set. However, the problem can be efficiently solved by linear programming techniques, applying a column and constraints generation method [1].

The uncertainty set D can be defined in different ways, depending on the available information: link load measurements and historical routing, a set of previously estimated TMs $\{\mathbf{d}^1, \mathbf{d}^2, \ldots, \mathbf{d}^o\}$ (average or peak values of traffic in the past), TM time series $\mathbf{d}(t)$, etc. We present an analysis of SRR advantages over traditional routing methods, considering two different definitions for D. The study is performed 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 [9]. Traffic data consists in 6-month traffic matrices collected every 5' via Netflow from the Abilene Observatory [8].

2.2 Robust routing with instantaneous traffic measurements

We first consider the problem of TM estimation. Let R_o be the historical routing matrix of Abilene, not necessarily optimal (R_o is available at [9]). Given a single instantaneous scenario (e.g. a 5' sample), we consider the current links' load vector $\mathbf{y}_o$. We define the uncertainty set D as all the TMs which are consistent with routing and link load measurements:

$$D = \{\mathbf{d} \in \mathbb{R}^m, R_o \times \mathbf{d} = \mathbf{y}_o,\ \mathbf{d} \geqslant 0\} \tag{10}$$

We compare the traditional and SRR approaches considering three different scenarios: *ideal scenario:* real traffic demand $\mathbf{d}^*$ is completely known; *traditional scenario:* traffic demand is estimated (TM estimations are available in [9], we consider the tomogravity estimation [6]); *robust scenario:* all we know is that $\mathbf{d}^*$ belongs to D. In the *ideal scenario*, routing is optimized for the real traffic demand $\mathbf{d}^*$, obtaining a MLU u^*_{max}. Both routing configuration and u^*_{max} are the solution for the traditional optimization problem (3). In the *traditional scenario*, routing configuration is optimized for an estimated traffic demand $\hat{\mathbf{d}}$. This routing configuration is the solution of problem (3), when demand's value $\mathbf{d}$ is $\hat{\mathbf{d}}$. Let us call this routing configuration $\hat{R}$. We use $\hat{R}$ to route the real demand $\mathbf{d}^*$, obtaining a value of MLU $\hat{u}_{max}$ (according to (2), $\hat{u}_{max} = u_{max}(C, \mathbf{d}^*, \hat{R})$). The reader should note that real traffic matrix $\mathbf{d}^*$ could be in fact any point of D. In this sense, we compute the worst case MLU for the estimated routing configuration $\hat{R}$ within the uncertainty set D:

$$\hat{u}^{wc}_{max} = \max_{\mathbf{d} \in D} \ u_{max}\left(C, \mathbf{d}, \hat{R}\right)$$

In the *robust scenario*, a robust routing configuration is computed for D, according to problem (8). The obtained routing configuration and MLU are called R_{robust} and $u^{robust\ wc}_{max}$ respectively. The value $u^{robust\ wc}_{max}$ represents the worst case performance for D (directly from (8), $u_{max}\left(C, \mathbf{d}, R_{robust}\right) \leqslant u^{robust\ wc}_{max}$, $\forall\ \mathbf{d} \in D$). Finally, we compute the MLU u^{robust}_{max} that results from routing the real demand $\mathbf{d}^*$ with R_{robust}:

$$u^{robust}_{max} = u_{max}\left(C, \mathbf{d}^*, R_{robust}\right)$$

We repeat the same evaluation for different times of the day. For each of them, ideal, traditional and robust routing performances are compared. Table 2 summarizes the results of this comparison. For simplicity, we take $u^*_{max} = 1$ as reference. Let us consider the obtained results for the 14:00 time of day (column 14:00). If the value of the traffic demand were known, the MLU would be u^*_{max}. In practice, it is difficult to perfectly know the value of the traffic demand, so an estimation is used. If the routing is optimized for the estimated value $\hat{\mathbf{d}}$ (*traditional scenario*), then the performance of that routing $\hat{R}$ when the traffic demand value is $\mathbf{d}^*$ is $1.07u^*_{max}$. Thus, the performance degradation due to the estimation is 7%, which is reasonable provided that the links' utilization is not too large. This means that the estimate is sufficiently close to the true value of the demand to make load balancing possible and efficient (at least in the case of the Abilene dataset [9]). But in theory, the only thing we can be sure of is that $\mathbf{d}^*$ belongs to the uncertainty set D and nothing proves with certainty that $\mathbf{d}^*$ is *close* to the estimated value $\hat{\mathbf{d}}$. If we take into consideration that the traffic demand takes any value in D, then the MLU can reach $5.75u^*_{max}$ in the worst case and this is obviously a risk that nobody would be ready to take.

Day Time	**02:00**	**08:00**	**14:00**	**20:00**
$\hat{u}_{max}$	1.18	1.03	1.07	1.07
u^{robust}_{max}	1.07	1.14	1.15	1.13
$\hat{u}^{wc}_{max}$	4.71	4.87	5.75	5.01
$u^{robust\ wc}_{max}$	1.10	1.15	1.16	1.14

Table 2: Routing performance under traffic uncertainty, relative to u^*_{max}.

Now let us suppose that this uncertainty is taken into account preventively in the optimization of the routing (*robust scenario*). In that case, the MLU when the traffic demand value is $\mathbf{d}^*$ is $1.15u^*_{max}$; compared to the performance of the traditional approach, the robust routing "cost" is $1.15 - 1.07 = 0.08$, i.e. a 8% performance degradation. But the MLU in the SRR case will always be bounded by $1.16u^*_{max}$, whatever the value of $\mathbf{d}$ in D. Compared to the $5.75u^*_{max}$ worst case performance of the traditional approach, it is clear that the robust approach offers a guarantee against the uncertainty on the traffic demand value, for a cost which remains reasonable (8%).

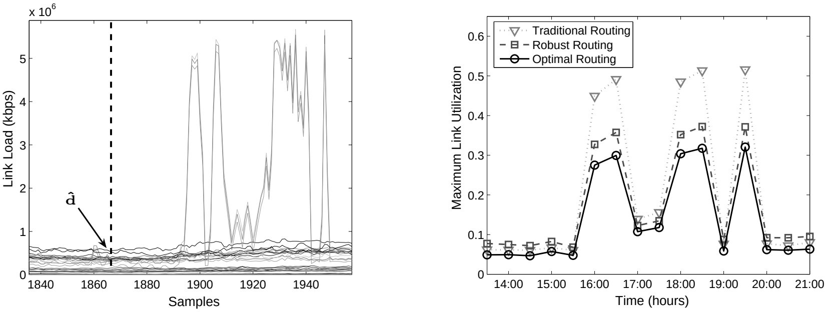
2.3 Robust routing with time-varying demands

Robust optimization can also be used to handle time-varying demands. In this case, the uncertainty set will be defined on the basis of historical traffic information, bounding all possible time-realizations of link loads by some upper-bound $\mathbf{y}_{UB}$:

$$D = \{\mathbf{d} \in \mathbb{R}^p, R_o \times \mathbf{d} \leqslant \mathbf{y}_{UB},\ \mathbf{d} \geqslant 0\} \tag{11}$$

An obvious example for this upper-bound would be the link capacities, $\mathbf{y}_{UB} = C$. A more interesting upper-bound could be the peak-usage-hour traffic load $\mathbf{y}_{UB} = \mathbf{y}_{peak}$, or even more, some *preventive threshold* could be considered for unexpected traffic variations $\mathbf{y}_{UB} = \beta.\mathbf{y}_{peak}$, with $\beta \geqslant 1$.

We compare once again the traditional and robust routing approaches, but taking into account the considerations in 1.2. In this experience, we consider an unexpected abrupt change in link's load due to an external routing modification. Figure (1.a) presents this abrupt modification. We consider three different scenarios: in the *traditional scenario*, unexpected changes are not considered in advance. Single traffic matrix estimation is conducted (close to 14:00, see figure 1), and routing optimized for this estimation ($\hat{R}$) is then applied to the real traffic demand $\mathbf{d}^*$, during the whole evaluation period. In the *robust scenario*, an upper-bound based on peak links' load is considered ($\mathbf{y}_{UB} = \mathbf{y}_{max}^{00:00\ -\ 24:00}$). Routing is optimized for this uncertainty set (R_{robust}) and is then applied to route real traffic demands. Both approaches are compared to the *ideal scenario*, where real traffic is completely known and routing optimization is performed at each time interval (*an optimal dynamic routing*).



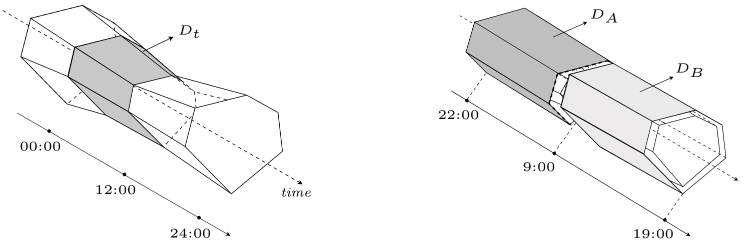
(a) (b)

Figure 1: (a) Daily traffic links' load, (b) Routing evaluation.

Figure (1.b) presents the routing performance evaluation. Before the abrupt change, traffic remains almost constant and MLU is similar for all scenarios, with the same slight differences as before. However, performance degradation for the traditional approach reaches approximately 60% after the arrival of the unexpected event, against a 10% degradation for SRR (both with respect to the ideal scenario). The traffic demand that is responsible for this abrupt modification in links' utilization belongs to the considered uncertainty set D, so the RR configuration is prepared to handle it.

3 MULTI-TEMPORAL ROBUST ROUTING

Previous analysis shows that the SRR approach offers stability guarantees against traffic uncertainty and traffic time-variations at a reasonable cost. However, we show that considering a single routing scheme for long-time periods is conservative and results in sub-optimal performance. We propose a simple approach to adapt the uncertainty set that outperforms the SRR. Based on rough knowledge of traffic variations (considering expected traffic behaviour), we propose to divide the uncertainty set and build a multi-temporal routing configuration, considering a single SRR configuration for each sub-set.



(a) (b)

Figure 2: (a) Daily variation of the polytope, (b) Time partitioning of the polytope.

Daily traffic changes can be seen as a time variation of the uncertainty set. At each time interval t_j, the routing matrix R and the link load values $\mathbf{y}^{t_j}$ define an instantaneous uncertainty set $D_{t_j} = \left\{\mathbf{d} \in \mathbb{R}^m, R \times \mathbf{d} \leqslant \mathbf{y}^{t_j},\ \mathbf{d} \geqslant 0\right\}$. The union of several instantaneous uncertainty sets along time t defines the *daily uncertainty set* $D_t = \left\{(\mathbf{d}, t) \in \mathbb{R}^{m+1}, \mathbf{d} \in \cup_{j=1}^{\tau} D_{t_j},\ t_1 \leqslant t \leqslant t_\tau\right\}$.

Figure (2.a) explains this idea. Assuming this set is the union of several polytopes, [5] provides a theoretical study of the optimal partitioning of D_t, using a partitioning hyper plane. [5] proves that this is a NP-hard problem, except for the case where a partitioning direction is previously fixed.

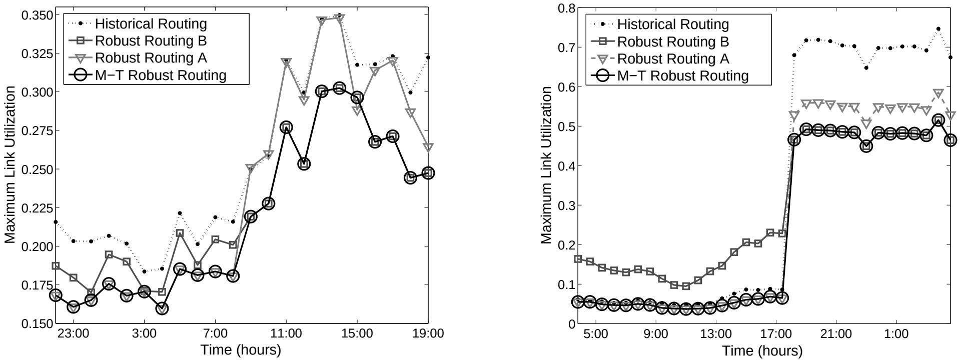
We define a partitioning hyper plane by its direction vector α and a value w: $\alpha.\mathbf{d} = w$. In the MTRR approach, we consider a particular direction for partitioning: the *time direction.* In that case, w represents the time of the day. We define $s+1$ hyper planes at times $\{w_1, w_2, .., w_{s+1}\}$. The intersection between D_t and the half-spaces defined by these partitioning hyper planes results in s uncertainty sub-sets $D_i = \{D_t \cap \{\mathbf{d}, \alpha.\mathbf{d} \geqslant w_i\} \cap \{\mathbf{d}, \alpha.\mathbf{d} \leqslant w_{i+1}\}\}, \forall i = 1, .., s.$ A SRR configuration R^i_{robust} is computed for each sub set D_i. Each routing configuration is finally applied at each time interval. The optimal values of routing changes $\mathbf{w}^* = \{w^*_2, \ldots, w^*_s\}$ are the solution for the following optimization problem (w_1 and w_{s+1} are fixed a priori):

$$\mathbf{w}^*(D_t) = \arg \min_{\mathbf{w}} \left\{ \max_{i=1..s} \; u_{max}(D_i) \right\} \tag{12}$$

where $u_{max}(D_i)$ is the solution of (8) for polytope D_i. [5] presents a simple algorithm to approximately solve (12), using a dichotomy methodology. The MTRR 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.

4 MTRR EVALUATION

We present a comparative analysis between SRR and MTRR in the same previously used network topology [9]. Abilene nodes are distributed over different time zones and therefore, links' traffic is not synchronized. Said in other words, the time-variation of the polytope is not a simple homothety, so a routing configuration change during the day can improve routing performance. We consider a single time partitioning (i.e. 2 routing intervals), $w_1 =$ 22:00, $w_2 =$ 9:00 and $w_3 =$ 19:00. For each time interval, we consider the smallest polytope that includes all possible realizations over that period:



(a) (b)

Figure 3: Routing performance, stable vs. multi-temporal robust routing.

$$D_{A,B} = \{\mathbf{d} \in \mathbb{R}^m, R_o \times \mathbf{d} \leqslant \mathbf{y}_{A,B}, \ \mathbf{d} \geqslant 0\}$$

where R_o is the historical routing matrix of Abilene, $\mathbf{y}_A = \mathbf{y}^{22:00-9:00}_{max}$ and $\mathbf{y}_B = \mathbf{y}^{9:00-19:00}_{max}$ (maximum values for each link). In this way, D_A includes all traffic demands between 22:00

and 9:00 and D_B between 9:00 and 19:00. Figure (2.b) shows these two polytopes with respect to the daily uncertainty set D_t. For each polytope, we compute a SRR configuration, R^A_{robust} and R^B_{robust}. In order to compare stable and multi-temporal approaches, we apply both routing configurations during the whole evaluation period.

Figure (3.a) compares the routing performance (MLU) between these two RR configurations. The obtained results show that the polytope D_A is well suited for smaller loads (see the *historical routing* curve), so R^A_{robust} 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^B_{robust}. The MTRR consists in computing the moment when routing must be changed (w^*_2 = 9:00 in this case), using the corresponding routing configuration depending on the time of the day (R^A_{robust} before w^*_2 and R^B_{robust} after). In this experience, the MTRR approach presents a performance improvement of almost 15% with respect to the SRR approach during the evaluation period. We now consider traffic demands that drastically change (i.e. a large time-variation of the polytope, caused by an unexpected event). Figure (3.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^* = 18:00. The MTRR approach definitely outperforms the SRR in this experience, presenting a MLU between 10% and 60% smaller during the whole evaluation period.

5 CONCLUSIONS AND FUTURE WORK

This paper addressed the routing optimization under traffic uncertainty problem. We presented a general overview of the different traffic engineering approaches for routing optimization, analyzing the tradeoffs between stable and dynamic routing and discussing their weaknesses facing current highly variable traffic demands. We explored the robust routing paradigm, an approach that considers uncertainty within the routing optimization to enhance reliability. Using a data set from a real network, we showed that this routing approach achieves robustness at low optimality loss in different situations. We introduced a simple multi-temporal robust routing approach that outperforms previous stable robust routing when considering traffic time variations. Taking advantage of traffic patterns, we split the uncertainty set into different time-consecutive sub-sets and compute a stable robust routing configuration for each of them. We believe that this approach represents a first step towards a dynamic and robust routing policy.

This preliminary work does not evaluate the challenges of changing routing configuration and its impact on end-to-end traffic. The use of an automatic methodology for detecting traffic variations and updating the robust routing configuration should also be explored, considering the application of the MTRR in a real network deployment.

REFERENCES

[1] W. Ben-Ameur and H. Kerivin, "Routing of Uncertain Traffic Demands", in *Optimization and Engineering*, 6, 283-313, 2005.

[2] D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs", in *Proc. ACM SIGCOMM*, 2003.

[3] M. Johansson and A. Gunnar, "Data-driven Traffic Engineering: techniques, experiences and challenges", in *BROADNETS 2006*, 2006.

[4] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang and A. Greenberg, "COPE: Traffic Engineering in Dynamic Networks", in *Proc. ACM SIGCOMM*, 2006.

[5] W. Ben-Ameur, "Between Fully Dynamic Routing and Robust Stable Routing", in *DRCN 2007*, 2007.

[6] A. Medina, K. Salamatian, S. Bhattacharyya and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *Proc. ACM SIGCOMM*, 2002.

[7] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford and F. True, "Deriving Traffic Demands for Operational IP Networks", in *IEEE/ACM Trans. on Networking*, vol. 9, no. 3, pp. 265-279, 2001.

[8] The Abilene Observatory, *http://abilene.internet2.edu/observatory/*.

[9] Y. Zhang, "Abilene Dataset 04", *http://www.cs.utexas.edu/yzhang/research/AbileneTM/*.