

Robust Routing vs Dynamic Load-Balancing

A Comprehensive Study and New Directions

Pedro Casas^{*‡}, Federico Larroca[†], Jean-Louis Rougier[†] and Sandrine Vaton^{*}

^{*}Télécom Bretagne, Brest, France - Email: name.surname@telecom-bretagne.eu

[†]Télécom ParisTech, Paris, France - Email: name.surname@telecom-paristech.fr

[‡]Universidad de la Repùblica, Montevideo, Uruguay

Abstract—Traffic Engineering (TE) has become a challenging task for network management and resources optimization due to traffic uncertainty and to the difficulty to predict traffic variations. To address this uncertainty in a robust and efficient way, two almost antagonist approaches have emerged during the last years: Robust Routing and Dynamic Load-Balancing. The former copes with traffic uncertainty in an off-line preemptive fashion, computing a stable routing configuration that is optimized for a large set of possible traffic demands. The latter balances traffic among multiple paths in an on-line reactive fashion, adapting to traffic variations in order to optimize a certain cost-function. Much has been said and discussed about the advantages and drawbacks of each approach, but very few works have tried to compare the performance of both mechanisms, particularly in the same network and traffic scenarios. This paper brings insight into several Robust Routing and Dynamic Load-Balancing algorithms, evaluating their virtues and shortcomings and presenting new mechanisms to improve previous proposals. Among others, such a study intends to help network operators in choosing an adequate mechanism to cope with traffic uncertainty.

Index Terms—Traffic Uncertainty, Stable and Reactive Robust Routing, Dynamic Load Balancing.

I. INTRODUCTION

As network services and Internet applications evolve, network traffic is becoming increasingly complex and dynamic. The convergence of data, telephony and television services on an all-IP network directly translates into a much higher variability and complexity of the traffic injected into the network. Recent Internet traffic studies from major network technology vendors like Cisco Systems forecast the advent of the Exabyte era [1], a massive increase in network traffic driven by high-definition video. Furthermore, current evolution and deployment-rate of broadband access technologies (e.g. Fiber To The Home) is such that the assumption of infinitely provisioned core links will soon become obsolete. Thus, simply upgrading link capacities may no longer be an economically viable solution. To make matters worse, the presence of unexpected events such as network equipment failures, large-volume network attacks, flash crowd occurrences and even external routing modifications induces large uncertainty in traffic patterns. In the light of this traffic scenario, network operators are searching for reliable routing mechanisms. The term *reliability* refers to the ability of the network to perform

and maintain its functions in usual circumstances, as well as under unexpected events. Two almost antagonist approaches have emerged in the recent years to cope with both the traffic increasing dynamism and the need for cost-effective solutions: Robust Routing (RR) and Dynamic Load-Balancing (DLB).

In RR, traffic uncertainty is taken into account directly within the routing optimization, computing a single routing configuration for all traffic demands within some *uncertainty set* where traffic is assumed to vary. This uncertainty set can be defined in different ways, depending on the available information: largest values of links load previously seen, a set of previously observed demands (previous day, same day of the previous week, etc.), etc. The criteria to search for this unique routing configuration is generally to minimize the maximum link utilization over all demands of the corresponding uncertainty set. While this routing configuration is not optimal for any single demand within the set, it minimizes the worst case performance over the whole set.

DLB copes with traffic uncertainty and variability by splitting traffic among multiple paths in real-time. In this dynamic scheme, each origin-destination pair of nodes within the network is connected by several a priori configured paths, and the problem is simply how to distribute traffic among these paths in order to optimize a certain function. DLB is generally defined in terms of a link-cost function, where the portions of traffic are adjusted in order to minimize the total network cost.

Those who promote DLB highlight among others the fact that it is the most resource-efficient possible scheme, and that given the configured paths it supports every possible traffic demand, all of this in an automated and decentralized fashion. Those who advocate the use of RR claim that there is actually no need to implement supposedly complicated dynamic routing mechanisms, and that the incurred performance loss for using a single routing configuration is negligible when compared with the increase in complexity. Network operators are reluctant to use dynamic mechanisms and prefer stable routing configurations, as they claim they get a better feeling of what is going on in the network. In this work we provide substantial evidence on the virtues and shortcomings of both mechanisms, on the one hand by quantifying the performance loss of RR with respect to DLB, and on the other hand by analyzing the temporal response of RR and DLB under significant and unpredicted traffic variations.

A. Related Work

There is a large literature on routing optimization with uncertain traffic demands. Thus, here we shall only mention a few papers, and do not intend our list to be exhaustive.

Traditional routing optimization algorithms rely on a single or a small group of expected traffic demands [2], [3]. In the scenario described before such approach is not suitable and Robust Routing techniques [4]–[6] should be used instead. The objective in Robust Routing is to find a unique static routing configuration that fulfills a certain criteria for a broad set of traffic demands, generally the one that minimizes the maximum link utilization over the whole set of demands. In [4], the authors capture traffic variations by introducing a polyhedral set of demands, which allows for easier and faster linear optimization. Oblivious Routing [5] also defines linear algorithms to optimize worst-case performance for different sizes of traffic uncertainty sets. The drawback of robust routing is its inherent dependence on the definition of the set of traffic demands: 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. Another problem related to robust routing is that optimization under uncertainty is generally more complex than classical optimization, which forces the use of simpler optimization criteria.

A completely different approach is provided by Dynamic Load-Balancing [7]–[9], where traffic is split among a priori established paths in order to avoid network congestion. Two of the most well-known proposals in this area are MATE and TeXCP. In MATE [7], a convex link cost function is defined, which depends on the link capacity and load. The objective is to minimize the sum over all links of this cost, for which a simple gradient descent method is proposed. TeXCP [8] proposes a somewhat simpler objective: minimize the biggest utilization each origin-destination pair obtains in its paths. A rough description of the algorithm is that origin nodes iteratively increase the portion of traffic sent through the path with the smallest utilization. Another load-balancing scheme which has the same objective but a relatively different mechanism is REPLEX [9]. Dynamic load-balancing presents a desirable property, that of keeping routing adapted to dynamic traffic. However, for these adaptive and distributed algorithms, a tradeoff between adaptivity (convergence speed) and stability must be found, which may be particularly difficult in situations where abrupt traffic changes may occur.

As regards a comparative study between RR and DLB, to the best of our knowledge the only paper that performs a similar analysis is [6]. This paper presents COPE, a RR mechanism that optimizes routing for predicted demands and bounds worst-case performance to ensure acceptable efficiency under unexpected traffic events. In this work, authors compare the performance of COPE with a dynamic approach which they claim models the behavior of mechanisms such as MATE and TeXCP. Given a time-series traffic demands, this dynamic approach consists of computing an optimal routing for each

traffic demand i and evaluate its performance with the following traffic demand $i+1$. There are two important shortcomings of this DLB simulation. Firstly, adaptation in DLB is iterative and never instantaneous. Secondly, in all DLB mechanisms paths are set a priori and remain unchanged during operation. This is not the case in their dynamic approach, where each new routing optimization may change not only traffic portions but paths themselves. For these reasons, we believe that the comparison provided in [6] is biased against dynamic schemes.

B. Contributions of the Paper

This paper presents a fair and comprehensive comparative analysis between RR and DLB mechanisms. The analysis is comprehensive as it evaluates the performance of both mechanisms based on different performance indicators and considering normal operation as well as unpredicted traffic events. We believe our comparison is fair because it considers the particular characteristics of each mechanisms under the same network and traffic conditions. To the date and to the best of our knowledge this is the first work that conducts such a comparative evaluation, necessary indeed not only from a research point of view but also for network operators who seek for proper solutions to face future network scenarios. Based on this comparative analysis we develop and evaluate new variants of RR and DLB mechanisms, improving some of the shortcomings found in both static and dynamic approaches.

The remainder of this paper is organized as follows. In Sec. II we introduce a preliminary version of the RR and DLB mechanisms, describe the framework of the evaluation, and discuss some first results. Sec. III presents new variants to the former mechanisms which alleviate the shortcomings detected. The evaluation of the complete set of algorithms under different traffic scenarios is conducted in Sec. IV. We finally draw conclusions of this comparative analysis in Sec. V.

II. STABLE ROBUST ROUTING AND DYNAMIC LOAD BALANCING

We will begin by introducing the notation used in this paper. The network topology is defined by n nodes and a set $L = \{l_1, \dots, l_q\}$ of q links, each with a corresponding capacity (c_1, c_2, \dots, c_q) . The Traffic Matrix (TM) $X = \{x_{i,j}\}$ denotes the traffic demand between every origin node i and every destination node j ($i \neq j$) of the network; we shall note each of these pairs as OD pairs. Let $X = \{x_k\}$ be the vector representation of the TM, where we have reordered OD pairs by index $k=1, \dots, m$ ($m=n.(n-1)$). Let $N = \{\text{OD}_1, \dots, \text{OD}_m\}$ be the set of m OD pairs. Each OD pair k can distribute its traffic arbitrarily among a set of w_k paths P_k . In this sense, we shall call r_p^k the portion of traffic x_k sent through path $p \in P_k$, where $0 \leq r_p^k \leq 1$ and $\sum_{p \in P_k} r_p^k = 1$.

Let λ_l^p be an indicator variable that is 1 if path p traverses link l and 0 otherwise, and $Y = \{\rho_1, \dots, \rho_q\}$ a vector representation of links load. Then X and Y are related through the routing matrix $R = \{r_l^k\}$, where $r_l^k = \sum_{p \in P_k} \lambda_l^p \cdot r_p^k$. The variable r_l^k indicates the fraction of traffic from OD pair k routed through link l . This results in the relation $Y = R \cdot X$.

Given X , the optimal multi-path routing problem consists in choosing the set of paths P_k for each OD pair k and computing the routing matrix R , in order to optimize a certain objective function $f(X, R)$. A simplified version of this problem is the optimal load-balancing problem which, given a set of paths, calculates R . In this work we consider different performance indicators, which result in different objective functions.

A very important link-level performance indicator is link utilization $u_l = \rho_l/c_l$; a value of u_l close to one indicates that the link is operating near its capacity. Network operators usually prefer to keep links utilization relatively low in order to support sudden traffic increases and link/node failures. A network-wide performance indicator is the maximum link utilization u_{\max} :

$$u_{\max}(X, R) = \max_{l \in L} \{u_l\} \quad (1)$$

The maximum link utilization constitutes by far the most popular TE objective function. However, its minimization presents a clear drawback: setting the focus too strictly on u_{\max} often leads to worse distributions of traffic, adversely affecting the mean network load and thus the end-to-end delay. In this sense we will consider the total network link utilization u_{tot} as another possible objective function:

$$u_{\text{tot}}(X, R) = \sum_{l \in L} u_l \quad (2)$$

Minimizing u_{tot} may provide better network-wide performance, as long as the maximum link utilization remains bounded; we discuss this issue in Sec. III.

The last performance indicator we shall consider in this work is the queuing delay. This indicator is quite versatile in the sense that big delays translate in bad performance for all traffic. Assume that queuing delay on link l is given by the function $d_l(\rho_l)$. Given this function we can compute the queuing delay of path p as $d_p = \sum_{l \in p} d_l(\rho_l)$. As a measure of the global performance, we consider the expected end-to-end queuing delay d_{mean} defined as:

$$d_{\text{mean}}(X, R) = \sum_{k \in N} \sum_{p \in P_k} (r_p^k \cdot x_k) d_p = \sum_{l \in L} \rho_l \cdot d_l(\rho_l) \quad (3)$$

That is to say, a weighted mean end-to-end queuing delay, where the weight for each path is how much traffic is sent through it ($r_p^k \cdot x_k$), or in terms of links, the weight for each link is how much traffic is traversing it (ρ_l). We prefer this performance indicator to a simple total delay because it reflects more precisely performance as perceived by traffic. Two situations where the total delay is the same, but in one of them most of the traffic is traversing heavily delayed links should not be considered as equivalent. Note that, by Little's law, the value $f_l(\rho_l) := \rho_l \cdot d_l(\rho_l)$ is proportional to the volume of data in the queue of link l . We will then use this last value as the addend in the last sum in (3), since it is easier to measure than the queuing delay.

Based on these definitions we will introduce the different optimization algorithms that strive to minimize some of these performance indicators, considering either the RR or DLB approach.

$\begin{aligned} & \text{minimize} && u_{\max} \\ & \text{subject to:} && \\ & \sum_{k \in N} \sum_{p \in P_k} \lambda_l^p \cdot r_p^k \cdot x_k \leq u_{\max} \cdot c_l && \forall l \in L, \forall X \in \mathbb{X} \\ & \sum_{p \in P(k)} r_p^k = 1 && \forall k \in N \\ & r_p^k \geq 0 && \forall p \in P_k, \forall k \in N \\ & u_{\max} \leq 1 && \end{aligned}$
--

TABLE I
ROBUST ROUTING OPTIMIZATION PROBLEM (RROP)

A. Stable Robust Routing

In a robust perspective of TE, demand uncertainty is preemptively taken into account within the routing optimization, computing a single routing configuration for all demands within some *uncertainty set*. In this work we consider a polyhedral uncertainty set \mathbb{X} , more precisely a *polytope* as in [4], based on the intersection of several half-spaces that result from linear constraints imposed to traffic demand. As an example, let us define an uncertainty set \mathbb{X} based on a given routing matrix R_o and the peak-hour links traffic load Y^{peak} obtained with this routing matrix:

$$\mathbb{X} = \{X \in \mathbb{R}^m, R_o \cdot X \leq Y^{\text{peak}}, X \geq 0\}$$

This definition of the uncertainty set has a major advantage: routing optimization can be performed from easily available links traffic load Y without even knowing the actual value of traffic demand X .

The now traditional Robust Routing Optimization Problem (RROP) [4] consists on minimizing the maximum link utilization u_{\max} , considering all demands within polytope \mathbb{X} (see Table I). The solution to the problem is twofold; on the one hand, a routing configuration $R_{\text{robust}} = \operatorname{argmin}_R \max_{X \in \mathbb{X}} u_{\max}(X, R)$ and on the other hand, a worst-case performance threshold $u_{\max}^{\text{robust}} = \max_{X \in \mathbb{X}} u_{\max}(X, R)$. Given a proper definition of the uncertainty set, the obtained robust routing configuration is applied during long-term periods of time; in this sense, we refer to robust routing as *Stable Robust Routing* (SRR).

The authors of [4] have shown that RROP can be efficiently solved by linear programming techniques, applying a combined columns and constraints generation method. This method iteratively solves the problem, progressively adding new constraints and new columns to the problem. The new constraints are the extreme points of the uncertainty set \mathbb{X} , and the new columns represent new paths added to reduce the objective function value. Only extreme points of \mathbb{X} are added as new constraints, as it is easy to see that every traffic demand $X \in \mathbb{X}$ can be expressed as a linear combination of these extreme demands.

Regarding new added paths, the algorithm in [4] may not be the best choice from a practical point of view since the number of paths for each OD pair is not a priori restricted and the characteristics of added paths are not controlled. For example, it would be interesting to have disjoint paths to route

traffic from each single OD pair, improving resilience. For this reason we modify the algorithm to select new paths, both limiting the maximum number of paths in P_k and taking as new candidates the shortest paths with respect to link weights $w_l(i) = (\epsilon + (1 - r_l^k(i)))^{-1}$. In this case, $r_l^k(i)$ corresponds to the fraction of traffic x_k that traverses link l after iteration i and ϵ is a small constant that avoids numerical problems. If OD pair k uses a single path p , $r_l^k = 1$ for every link $l \in p$, and so this path is removed from the graph where new shortest paths are computed ($w_l \rightarrow \infty, \forall l \in p$). While this may result in a sub-optimal performance, it allows a real and practical implementation. In case there are no disjoint paths for OD pair k , we use the column constraint generation method used in [4] to add new paths for OD pair k .

B. Dynamic Load Balancing

As mentioned before, the objective in DLB is to minimize a certain objective function $f(X, R)$ in a distributed fashion (i.e. without relying on any centralized entity). Algorithms that achieve this are typically greedy, which present the desirable property of requiring minimum coordination among OD pairs. In this kind of mechanisms, a path cost ϕ_p is defined, and each OD pair greedily minimizes the cost it obtains from each of its paths. This context constitutes an ideal case study for game theory, and is known as *Routing Game* in its lingo [10], [11].

Since each OD pair may arbitrarily balance traffic among its paths, we will assume that OD pairs are constituted of infinitely many agents. These agents control an infinitesimal amount of traffic, and decide through which path to send their traffic. In this context r_p^k then represents the portion of agents of OD pair k that have p as their choice. If each of these agents acts selfishly, then the system will be at equilibrium when no agent may decrease its cost by changing its path decision. This situation constitutes what is known as a *Wardrop Equilibrium* (WE) [12], which is formally defined as follows:

Definition 1: The paths vector $\{r_p^k\}_{p \in P_k}$ is a Wardrop Equilibrium if for each OD pair $k \in N$ and for each couple of paths $p, q \in P_k$ with $r_p^k > 0$ it holds that $\phi_p \leq \phi_q$.

The path cost ϕ_p is in turn defined in terms of a link cost function $\phi_l(\rho_l)$. There are roughly two kinds of games depending on the definition of ϕ_p . A *Bottleneck Routing Game* defines $\phi_p = \max_{l \in p} \phi_l(\rho_l)$. It may be proved that its WE coincides with the paths vector that minimizes the maximum link cost (i.e. $R = \operatorname{argmin}_R \max_l \phi_l(\rho_l)$) [13]. It is easy then to see that if we were interested in minimizing $u_{\max}(X, R)$, the WE of a Bottleneck Routing Game with $\phi_l(\rho_l) = u_l$ coincides with the optimum. In the rest of the paper we shall note this game as MinUG (Minimum Utilization Game).

A *Congestion Routing Game* on the other hand, defines the path cost as $\phi_p = \sum_{l \in p} \phi_l(\rho_l)$. The WE of a Congestion Routing Game coincides with a local minimum of the potential function $\Phi(R) = \sum_{l \in L} \int_0^{\rho_l} \phi_l(x) dx$ [10]. If we were to minimize $d_{\text{mean}}(X, R)$, we should then play a Congestion Routing Game with a link cost equal to the derivative of the link mean queue size (i.e. $\phi_l(\rho_l) = f'_l(\rho_l)$). We shall note this

second game as MinDG (Minimum Delay Game).

We will now briefly discuss how, given the path cost function ϕ_p , the WE may be achieved for both routing games. In a recent article [14], the authors proved that if all OD pairs use *no-regret* algorithms, and under certain simple conditions on ϕ_p (positive, continuous and non-decreasing), the global behavior will approach the WE. One such algorithm is the classic *Weighted Majority Algorithm* (WMA) [15], whose pseudo-code for OD pair k is the following:

Algorithm 1 Weighted Majority Algorithm (WMA)

```

1: for  $t = 1, \dots, \infty$  do
2:   Obtain path costs  $\phi_p \forall p \in P_k$ 
3:   for every path  $p \in P_k$  do
4:     if  $\phi_p > \min_{q \in P_k} \phi_q$  then
5:        $r_p^k \leftarrow \beta \times r_p^k$ 
6:     end if
7:   end for
8:   Normalize the  $r_p^k$ 
9: end for
```

At each iteration t , those paths whose cost is bigger than the minimum are punished by multiplying their respective r_p^k by a certain constant $\beta < 1$ (throughout our simulations we have used $\beta = 0.95$). Actually, and in order to avoid unnecessary changes in the traffic distribution, we shall only update r_p^k when the corresponding path cost is bigger than the minimum cost plus a certain margin (in the case of MinUG we fixed the margin at 0.005, and for MinDG we used 5% of the minimum).

C. A Preliminary Comparison

In this subsection we shall present some first simulations that will help us gain insight into the mechanisms and highlight some of their respective shortcomings. Before, we will discuss how we performed these and the rest of the simulations.

As the reference network we used Abilene, an Internet2 backbone network. Abilene consists of 12 router-level nodes connected by 30 links (only intra-domain links were considered). The used topology and traffic demands are available at [18]. Traffic data consists of 6 months of traffic matrices collected every 5 minutes via Netflow. As measured demands do not significantly load the network, we re-scaled them by multiplying all their entries by a constant. The function $f_l(\rho_l)$ and its derivative were obtained as described in [16]. The method described in the paper allows to obtain an accurate approximation of the actual function $f_l(\rho_l)$ from link load and queue size measurements without assuming any given model. It is important to mention that in [16] we show that a simple M/M/1 model has little to do with real measurements.

To be as fair as possible, all mechanisms use the same set of paths, namely those calculated by SRR as discussed in Sec. II-A. The polytope \mathbb{X} is computed based on the historical routing matrix of Abilene R_o : $\mathbb{X} = \{X \in \mathbb{R}^m, R_o \cdot X \leq Y^{\max}, X \geq 0\}$, where Y^{\max} contains the maximum link load values observed at normal operation during the evaluation period. R_o is available at [18].

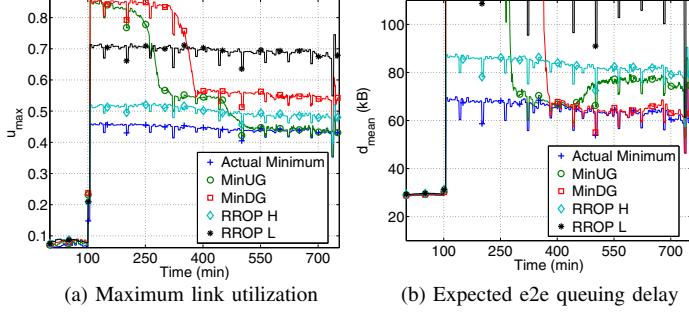


Fig. 1. Maximum link utilization and expected end-to-end queuing delay over time.

The TMs are fed to the mechanisms in consecutive temporal order. Both DLB mechanisms (MinUG and MinDG) are initiated at arbitrary values of r_p^k , which will be updated as new link load measurements arrive. We have assumed that each OD pair receives these measurements every minute, meaning that for each new TM five updates of their corresponding r_p^k values will be performed. Results are shown then for every minute. As a reference, we also computed the optimum values $u_{\max}(X, R)$ and $d_{\text{mean}}(X, R)$ for every TM X of the dataset.

Figure 1(a) shows the results for the maximum link utilization u_{\max} . The considered demands are the TMs with indexes between 1050 and 1200 from dataset X23 in [18]. The evaluation starts with a normal traffic pattern, where u_{\max} does not even exceed 0.10. At the 100th minute one of the OD pairs abruptly starts generating an anomalous amount of traffic. We consider two different definitions for polytope \mathbb{X} in the evaluation of SRR: a first polytope adapted to low load traffic, i.e. traffic in normal operation before the 100th minute, and a second polytope adapted to high load traffic, i.e. traffic after the occurrence of the anomaly. In the sequel we refer to SRR as RROP, recalling that the optimization problem is the one in Table I. In this sense, we shall use RROP L for the SRR adapted to normal operation traffic and RROP H for the SRR adapted to high load anomalous traffic. There is a major difference between RROP H and RROP L; in the first case, the anomalous traffic belongs to the uncertainty set and results are only 0.05 higher than the optimum. RROP L performs quite bad during the anomalous event, a result somehow expected given the definition of the polytope.

On the other hand, dynamic schemes have an important overshoot, with a difference with the optimum u_{\max} of approximately 0.40. Moreover, it takes them more than 200 minutes to converge. However, it should be noted that when it eventually converges, MinUG obtains a maximum utilization very similar to the optimum. Since it was not designed with this performance indicator in mind, MinDG obtains a difference with respect to the optimum, that in this case is approximately 0.10. Similar results are obtained for d_{mean} in Fig. 1(b). Once MinDG converges, it obtains an end-to-end queuing delay very similar to the actual optimum. On the other hand, MinUG and RROP H obtains a queuing delay that exceeds the optimum by approximately 25%.

$$\begin{aligned}
 & \text{minimize} && u_{\text{aof}} = \alpha \cdot u_{\max} + (1 - \alpha) \cdot u_{\text{tot}} \\
 & \text{subject to:} && \\
 & \sum_{l \in L} \sum_{k \in N} \sum_{p \in P_k} \frac{\lambda_l^p}{c_l} r_p^k \cdot x(k) \leq u_{\text{tot}} && \forall X \in \mathbb{X} \\
 & \sum_{k \in N} \sum_{p \in P_k} \lambda_l^p \cdot r_p^k \cdot x(k) \leq u_{\max} \cdot c_l && \forall l \in L, \forall X \in \mathbb{X} \\
 & \sum_{p \in P(k)} r_p^k = 1 && \forall k \in N \\
 & r_p^k \geq 0 && \forall p \in P_k, \forall k \in N
 \end{aligned}$$

TABLE II
ROBUST ROUTING AOF PROBLEM (RRAP)

III. IMPROVING THE ALGORITHMS PERFORMANCE

The simple evaluation conducted in Sec. II-C shows some conception drawbacks of both SRR and DLB algorithms presented in Sec. II. In this section we shall explain the origins of these problems and present enhanced mechanisms to overcome them.

A. Improving Stable Robust Routing

The SRR approach presents two important problems, one related to the objective function it intends to minimize and the other one as an inherent consequence of using a single static routing configuration. Let us begin by the objective function problem. As we stated in Sec. II, the minimization of u_{\max} may often lead to a worse distribution of traffic, adversely affecting for instance d_{mean} . This is exactly the case in Fig. 1(b) for RROP H. Using $d_{\text{mean}}(X, R)$ as the objective function in RROP results in a difficult optimization problem, as $f_l(\rho_l)$ is not a linear function. Instead, we could use the total network link utilization $u_{\text{tot}}(X, R)$ as the objective function. However, simply minimizing u_{tot} results in an unbounded value of u_{\max} , which is not practical from an operational point of view.

An alternative approach is to minimize both the value of u_{\max} and u_{tot} at the same time, which constitutes a multi-objective optimization problem. The difficulty that this kind of problem presents is that traditional single-objective optimization techniques cannot be directly applied. An intuitive and easy approach to address this issue is to construct a single aggregated objective function (AOF) that combines both objective functions. We define a weighted linear combination of u_{\max} and u_{tot} as the new objective function $u_{\text{aof}} = \alpha \cdot u_{\max} + (1 - \alpha) \cdot u_{\text{tot}}$, where $0 \leq \alpha \leq 1$ is the combination fraction. Table II presents the resulting optimization problem, which we shall note as Robust Routing AOF Problem (RRAP). The problem is solved using the same recursive algorithm as in RROP. As we show in the results from Sec. IV, the optimum of this problem achieves better global performance.

As we illustrated in Fig. 1, the other important drawback of SRR is its inherent dependence on the definition of the uncertainty set. In [17] we proposed an adaptive version of SRR, known as Reactive Robust Routing (RRR). The basic idea in RRR consists of computing a set of paths P_k and a robust routing configuration R_{robust}^o for expected traffic in

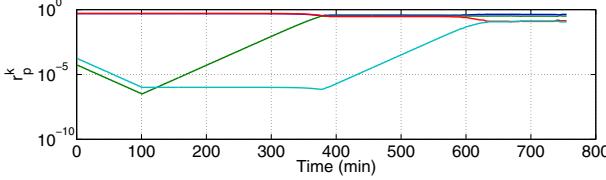


Fig. 2. Evolution of r_p^k for the anomalous OD pair (MinDG)

nominal operation. Additionally, and using this set of paths, we compute a routing configuration R_{robust}^j for every single anomalous traffic event A_j , $j \in N$, where A_j represents a large volume anomaly in traffic from OD pair j . Using a single anomaly detection/localization sequential algorithm, the RRR balances traffic among paths P_k according to R_{robust}^j when an anomaly of type A_j is detected and localized. This way we avoid to define an uncertainty set that encompasses all possible anomalous traffic. We refer the reader to [17] for details on the implementation of RRR.

B. Improving Dynamic Load Balancing

The DLB algorithms evaluated in Sec. II-C present an important overshoot and a significant settling time in the presence of sudden and large traffic variations. If the traffic anomaly is a perfect step, then the overshoot is unavoidable. We will try to address the long settling time instead. Figure 2 depicts the evolution over time of the corresponding r_p^k values of the anomalous OD pair for MinDG in the example of Sec. II-C. We may see that, although the r_p^k change exponentially fast, at the moment of the anomaly the values that should increase are so small that it takes them a very long time to converge. A possible solution is to impose a minimum value to all r_p^k . However, this will affect the precision of the algorithm and will still result in significant settling times.

Actually, r_p^k may be regarded as an indicator of the performance of path p in the previous iterations. A very small r_p^k means that p performed very badly with respect to the rest of the paths in the past. However, when the anomaly occurs, conditions severely change and history is no longer as relevant. If we consider that we are in such situation, we could for instance completely ignore history and restart the game by setting $r_p^k = 1/w_k \forall k \in P_k$. Before deciding how to reassign r_p^k , we will discuss how an OD pair may decide if it should restart its game or not.

Consider a situation where most of the traffic for OD pair k is routed through a path that is not the cheapest, and that the r_p^k corresponding to the minimum-cost path is very small. This could mean that although the former performed better in the past, this is no longer true and some traffic should be re-routed to the latter. This is more so as the difference in cost increases. However, this “suspicious” situation could be due to noisy measurements. To make sure that the game have actually changed and that it should be restarted, we will require such a situation to persists during a certain number of consecutive iterations. Once we detected that the game should be restarted, we will re-route some of the traffic that was being

routed through the path with the biggest r_p^k to the cheapest one. The amount will be proportional to the relative difference in cost to avoid overreacting. Finally, remember that with WMA fast adaptation is achieved when the r_p^k are not too small. The objective with this “game restart” is simply to move r_p^k from critically small values. The algorithm will then rapidly converge to the optimum. We now present the pseudo-code of the complete algorithm for OD pair k :

Algorithm 2 WMA with Restart (WMA-R)

```

1: for  $t = 1, \dots, \infty$  do
2:   Obtain path costs  $\phi_p \forall p \in P_k$ 
3:   Determine  $p_{\min} = \operatorname{argmin}_{p \in P_k} \phi_p$  and  $p_{\max} = \operatorname{argmax}_{p \in P_k} r_p^k$ 
4:   if ( $r_{p_{\min}}^k < 0.1$ ) and ( $\phi_{p_{\min}} + \phi_{th} < \phi_{p_{\max}}$ ) then
5:      $n_e^k \leftarrow n_e^k + 1$ 
6:   else
7:      $n_e^k \leftarrow 0$ 
8:   end if
9:   if  $n_e^k \leq n_{th}^k$  then
10:    Perform a normal iteration of WMA (cf. Algorithm 1)
11:   else
12:      $n_e^k \leftarrow 0$ 
13:      $\Delta_r \leftarrow \min \left\{ \frac{\phi_{p_{\max}}}{\phi_{p_{\min}}} - 1, 1 \right\} \times \frac{r_{p_{\max}}^k - r_{p_{\min}}^k}{2}$ 
14:      $r_{p_{\max}}^k \leftarrow r_{p_{\max}}^k - \Delta_r$ 
15:      $r_{p_{\min}}^k \leftarrow r_{p_{\min}}^k + \Delta_r$ 
16:   end if
17: end for

```

The new variable n_e^k counts the number of consecutive occurrences of a “suspicious” situation,. The threshold ϕ_{th} is to make sure that the difference in cost between paths is significant. In particular, we used $n_{th}^k = 3$, for MinUG we used $\phi_{th} = 0.005$ and for MinDG $\phi_{th} = 0.2\phi_{p_{\min}}$. Finally, note that when the game is restarted, we re-route a certain amount of traffic from p_{\max} to p_{\min} , but at most the amount of traffic routed through each path is equalized.

IV. EVALUATION AND DISCUSSION

In this section we evaluate the performance of the different RR and DLB algorithms presented in this work, considering both normal operation and anomalous traffic situations. We present and discuss three simulation case-scenarios: starting from a normal traffic variation scenario, we increase the number of OD pairs that present anomalous traffic variations. This allows for performance comparison at different levels of traffic variability.

A. Normal Operation

The first case-scenario is the simplest one, which corresponds to traffic in normal operation. The only variability is due to typical daily fluctuations. Fig. 3 shows the evolution over time of u_{\max} and d_{mean} for the different mechanisms, when fed with 260 TMs from set X01 in [18]. All algorithms perform similarly as regards maximum link utilization, depicted in Fig. 3(a). This may be further appreciated in Fig. 4(a), where we present a boxplot of the difference in u_{\max} with respect to the optimum for all the mechanisms.

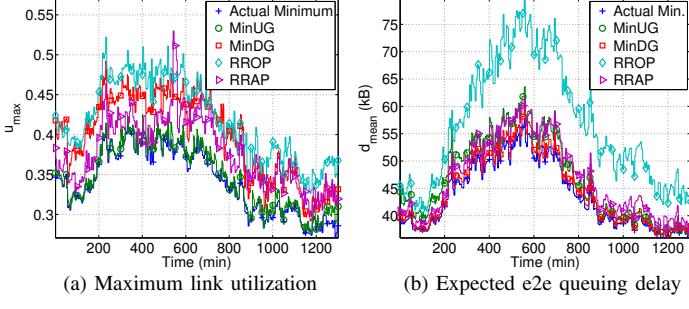


Fig. 3. Maximum link utilization and expected end-to-end queuing delay under normal operation.

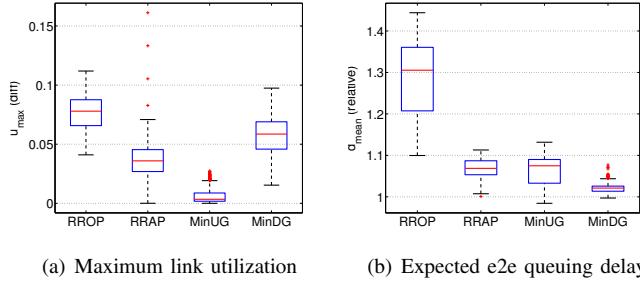


Fig. 4. Results overview for the first simulation case-scenario - normal operation.

Naturally, the algorithm that performs best is MinUG, although its improvement over the rest is not significant.

Results with respect to d_{mean} are quite different, as it can be seen in Fig. 3(b) and 4(b); the latter shows the d_{mean} obtained by each mechanism divided by the corresponding actual minimum. We may verify that the best results are obtained by MinDG, followed closely by both RRAP and MinUG. However, RROP systematically obtains a significant difference with respect to the optimum, generally of about 30%. These results further highlight the limitations of RROP as previously discussed in Sec. III-A: using only u_{max} as a performance objective results in a relatively low maximum utilization, but neglects the rest of the links, impacting global performance. For the case of RRAP, we used $\alpha = 0.5$ in this and the other case-scenarios.

B. One Anomalous OD Pair

The second case-scenario is the one considered in Sec. II-C, whose main characteristic was the sudden and abrupt increase of traffic generated by one OD pair. Notice in Fig. 5 how the improvements discussed in Sec. III-B for MinUG and MinDG result in a relatively smaller overshoot than before, but most importantly the settling time has been significantly decreased (cf. Fig. 1).

To be fair with DLB mechanisms, both RRAP and RROP use the RRR mechanism previously described in III-A to adapt traffic balancing after the detection of the anomalous traffic variation. Regarding u_{max} , Fig. 6(a) shows that results obtained by RRAP are quite good and close to the optimum. Surprisingly enough, the results for RROP are somewhat

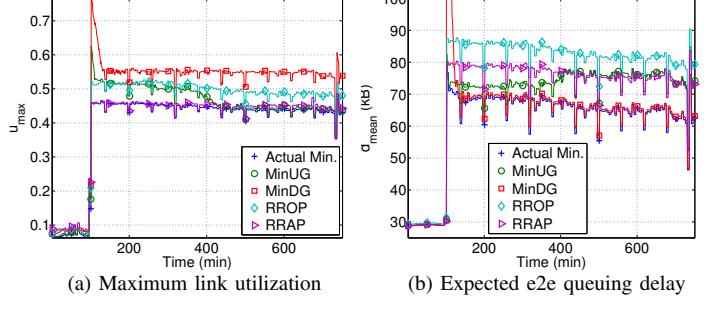


Fig. 5. Maximum link utilization and expected end-to-end queuing delay under abrupt and large traffic variations.

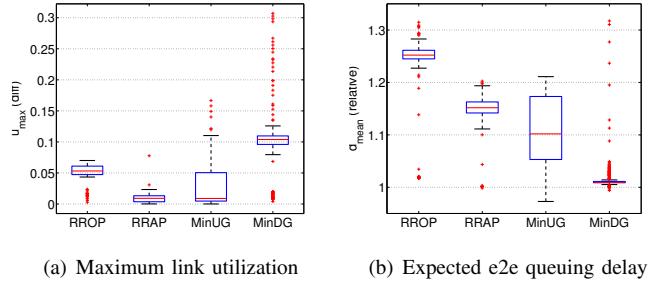


Fig. 6. Results overview for the second simulation case-scenario - abrupt traffic variation.

worse, although the difference is not significant. This difference is probably due to the fact that both RRAP and RRAP are solved numerically and not exactly in order to reduce computation time. Results for d_{mean} follow the same tendency as in Sec. IV-A, although both RRAP and MinUG perform somewhat worse than before.

C. Two Anomalous OD Pairs

In this case-scenario two OD pairs largely increase their traffic demand, one at approximately the 150th minute and the other at the 320th. They both present this anomalous traffic until the end of the simulation. We shall then separate the simulation in three parts: the first third where traffic is normal, the second third were only one OD pair is anomalous, and the last third were both OD pairs are anomalous. The anomaly localization algorithm of RRR was designed for the case of one single anomalous OD pair. Because of this, we will further illustrate the tradeoff between size of the considered uncertainty set and efficiency of the obtained routing, and chose the uncertainty polytope by the traffic loads seen after the second anomaly.

In Fig. 7(a) we may see that, as expected, the u_{max} obtained by both RRAP and RRAP in the last third of the simulation are very close to the optimum. However, in the rest of the simulation the difference may be important, specially in the second part where the difference for RRAP is almost 0.2. It is important to highlight the results obtained by MinDG and MinUG. Notice that the overshoot this time is much smaller than before (a maximum of 0.1 in u_{max} for MinDG) and the settling time is negligible. In this case, the increase in traffic

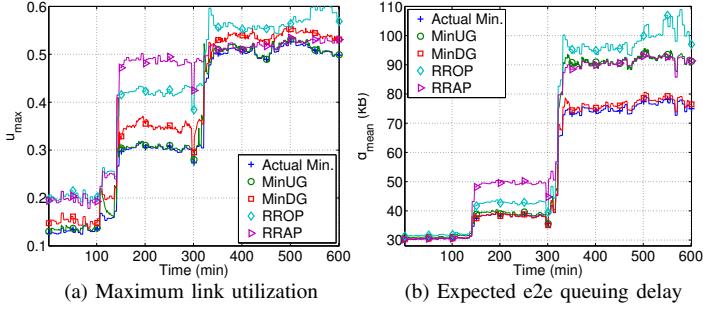


Fig. 7. Maximum link utilization and expected end-to-end queuing delay under gradual and large traffic variations.

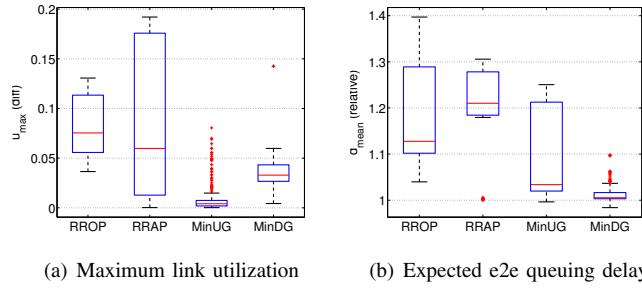


Fig. 8. Results overview for the third simulation case-scenario - gradual traffic variation.

of the anomalous OD pairs is more gradual than before, which clearly favors dynamic schemes in their performance.

V. CONCLUSIONS AND FUTURE WORK

From the study we performed in this paper we may reach several conclusions. The most important is probably that we have shown that using a single and unique routing configuration is not a viable solution when traffic is relatively dynamic. It obtains a very poor performance either when faced with unforeseen TMs or when its design tries to consider as many TMs as possible. It is clear from our study that some form of dynamism is necessary, which could be either RRR (Reactive Robust Routing) or DLB (Dynamic Load-Balancing).

RRR computes a nominal operation routing configuration, and has an alternative routing (using the same paths than in normal operation) for certain possible anomalous situations. In order to detect these anomalous situations, link load measurements have to be gathered. On the other hand, DLB gathers these same measurements but also requires updating load-balancing in a relatively small time-scale. The added complexity is then to distribute these measurements to all ingress routers (instead of a central entity) and updating the load-balancing in real-time.

Our results show that the additional complexity involved in DLB is not justified when the variability (or the anomalies) are not very significant. However, the use of DLB under highly dynamic traffic is very appealing and generally provides better results than RR. Moreover, if the anomalies may not be correctly detected, the only effective solution is DLB.

Regarding RR in particular, a local performance criterium such as u_{\max} , widely used in current network optimization problems, does not represent a suitable objective function as regards global network performance. The use of u_{\max} together with other performance indicator (such as u_{tot}) through the framework of Multi-Objective Optimization provides interesting results and deserves further analysis. For instance, we are currently working on the analysis of the Pareto-Optimal solutions of the problem, and comparing the resulting performance with that obtained by the AOF approach we presented here.

Dynamic approaches as DLB are generally met with reluctance due to their transient behavior under strong traffic variations. However, we have shown that this transient behavior can be effectively controlled, or at least alleviated, by simple mechanisms. Concerning the two different games we presented, conclusions are similar to those of RR. Minimizing d_{mean} instead of u_{\max} results in a somewhat bigger maximum utilization, but a (sometimes much) better global performance. It should also be highlighted that this paper represents one of the first studies using no-regret algorithms for load-balancing. The results presented here, which were obtained using the most basic of these algorithms, were very promising. There exist other more sophisticated algorithms of this kind, whose exploration we left for future work.

REFERENCES

- [1] Cisco-Systems, "Global IP Traffic Forecast and Methodology 2006-2011, white paper", 2007 - updated 2008, available at <http://www.cisco.com>.
- [2] M. Roughan, M. Thorup and Y. Zhang, "Traffic Engineering with Estimated Traffic Matrices", in *ACM IMC '03*, 2003.
- [3] C. Zhang, Z. Ge, J. Kurose, Y. Liu and D. Towsley, "On Optimal Routing with Multiple Traffic Matrices", in *IEEE INFOCOM '05*, 2005.
- [4] W. Ben-Ameur and H. Kerivin, "Routing of Uncertain Traffic Demands", in *Optimization and Engineering*, vol. 6, pp. 283-313, 2005.
- [5] D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs", in *ACM SIGCOMM '03*, 2003.
- [6] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang and A. Greenberg, "COPE: Traffic Engineering in Dynamic Networks", in *ACM SIGCOMM '06*, 2006.
- [7] A. Elwalid, C. Jin, S. Low and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", in *IEEE INFOCOM '01*, 2001.
- [8] S. Kandula, D. Katabi, B. Davie and A. Charny, "Walking the Tightrope: Responsive yet Stable Traffic Engineering", in *ACM SIGCOMM '05*, 2005.
- [9] S. Fischer, N. Kammenhuber and A. Feldmann, "REPLEX: dynamic traffic engineering based on wardrop routing policies", in *CoNEXT '06*, 2006.
- [10] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez and L. Wynter, "A survey on networking games in telecommunications", *Comput. Oper. Res.*, vol. 33, no. 2, pp. 286-311, 2006.
- [11] F. Larroca and J.L. Rougier, "Routing Games for Traffic Engineering", in *ICC '09*.
- [12] J. Wardrop, "Some theoretical aspects of road traffic research", *Proceedings of the Institution of Civil Engineers, Part II*, vol. 1, no. 36, pp. 352-362, 1952.
- [13] R. Banner and A. Orda, "Bottleneck Routing Games in Communication Networks", *IEEE Journal on Selected Areas in Comm.*, vol. 25, no. 6, pp. 1173-1179, 2007.
- [14] A. Blum, E. Even-Dar and K. Ligett, "Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games", in *PODC '06*.
- [15] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm", *Inf. Comput.*, vol. 108, no. 2, pp. 212-261, 1994.
- [16] F. Larroca and J.L. Rougier, "Minimum-Delay Load-Balancing Through Non-Parametric Regression", in *IFIP/TC6 Networking '09*, 2009.
- [17] P. Casas, L. Fillatre and S. Vaton, "Robust and Reactive Traffic Engineering for Dynamic Traffic Demands", in *NGI '08*, 2008.
- [18] Y. Zhang, "Abilene Dataset 04", <http://www.cs.utexas.edu/~yzhang/research/AbileneTM/>.