Adaptive Routing in IP Networks Using SNMP Link Counts

Josselin Vallet*†, Olivier Brun*‡
*CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse,France
†Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
‡Univ. de Toulouse, LAAS, F-31400 Toulouse, France
{jvallet, brun}@laas.fr

***Abstract*—The high volatility of traffic patterns in IP networks calls for dynamic routing schemes allowing to adapt resource utilization to prevailing traffic. In this paper, we focus on the problem of link weight optimization in OSPF networks where the traffic is routed along shortest paths according to the link metrics. We propose an online approach to optimize OSPF weights, and thus the routing paths, adaptively as some changes are observed in the traffic. The approach relies on the estimation of traffic demands using SNMP link counts. Experimental results on both simulated and real data show that the network congestion rate can be significantly reduced with respect to a static weight configuration.**

I. INTRODUCTION

The increasing popularity of bandwidth-hungry applications leads to more and more volatile traffic patterns. Designing a network using a "busy-hour" traffic matrix can lead to poor network performances. An alternative approach is to adapt resource utilization, by tailoring routes to prevailing traffic.

One difficulty is related to how traffic is routed by intra-domain routing protocols, the most prominent being Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) [1], [2]. Each traffic flow is routed along shortest paths, splitting equally the flow at nodes where several outgoing links are on shortest paths to the destination. Although usually set to one, the weights of the links can be changed by the network operator. The link weight optimization problem amounts to find links weights that optimize a given performance measure (see [3]–[5] and references therein). This problem is known to be NP-hard [6].

Traditional methods for link weight optimization have been designed for network planning and assume that a *predicted* traffic is known [7]–[10]. But traffic demands are difficult to measure precisely without an important overhead (ex: Netflow [11]), and even more more difficult to predict. We consider a lighter alternative approach using the SNMP (Simple Network Management Protocol) link counts (traffic on a link in a 5 min interval) to determine average demands. However, this problem cannot be solved exactly without additional information [12]–[17], and approximations have to be used.

In this paper, we investigate how to dynamically reconfigure link weights so as to adapt to prevailing traffic. Similar problem was considered in [18] but the authors assume to know the mean/variance the traffic, and in [19], where the authors fail to cope with real-time constraints. We propose an online algorithm for dynamic reconfiguration of IP routes depending on SNMP links loads, using both traffic estimation and robust optimization heuristic. Simulation results show that the proposed method allows to greatly improve network performances, while keeping running times compatible with an online execution.

The paper is organized as follows. Section II is devoted to the mathematical formulation of the problem. The proposed algorithm is described in Section III. Results obtained on both simulated and real data are presented in Section IV.

II. PROBLEM STATEMENT

The network is represented as a graph $G = (V, E)$. The sets V and E are composed of the N nodes and M links of the network, respectively. We denote by c_l the capacity of link l, and $K = N(N-1)$ the number of origin/destination (OD) pairs.

We observe the network at discrete time points $\tau = 1, 2, \ldots$ Let $\hat{\mathbf{y}}^\tau = (\hat{y}_1^\tau, \ldots, \hat{y}_M^\tau)$ be the vector of measured link traffics, where $\hat{y}_l^\tau$ gives the average traffic over link l in the time interval $\mathcal{I}_\tau = [\tau - 1, \tau]$. Let d_k^τ be the average traffic transmitted by OD pair k in $\mathcal{I}_\tau$. We denote the vector of traffic demands by $\mathbf{d}^\tau$.

Let $\boldsymbol{\omega}^\tau = (\omega_1^\tau, \ldots, \omega_M^\tau)$ be the vector describing the link weight configuration during $\mathcal{I}_\tau$, where the metric ω_l^τ of link l is an integer value in the interval $\Omega = \left[1, 2^{16} - 1\right]$. The $M \times K$ routing matrix $\mathbf{F}(\boldsymbol{\omega}^\tau)$ is obtained from $\boldsymbol{\omega}^\tau$ using any shortest path algorithm. Element $f_{l,k}(\boldsymbol{\omega}^\tau)$ is the fraction of demand k sent across link l.

Considering all this notations, the problem we address amounts to finding the weight configuration $\boldsymbol{\omega}^\tau$ that minimizes the congestion rate of the network (defined as the maximum utilization rate of the links):

$$\begin{aligned} \text{minimize} \quad & \rho(\boldsymbol{\omega}) = \max_{l \in E} \frac{y_l}{c_l} && \text{(METRIC)} \\ \text{subject to} \quad & \mathbf{y} = \mathbf{F}(\boldsymbol{\omega})\, \mathbf{d}^\tau, && (1) \\ & \boldsymbol{\omega} \in \Omega^M. \end{aligned}$$

We describe the proposed approach to solve this problem in the following section.

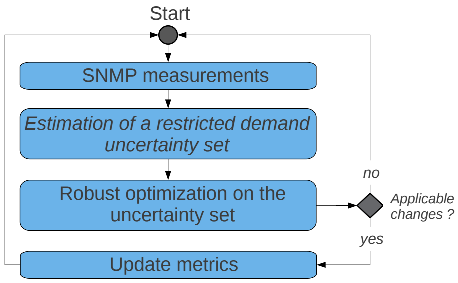

Fig. 1. Online algorithm

III. Online algorithm

The proposed online algorithm for dynamic reconfiguration of IP routes is described in Figure 1. This algorithm is run periodically. It first uses SNMP to collect the average traffic on each link over the last time window. These observations are used to estimate a demand uncertainty set, over which the routing metrics have to be optimized. A robust greedy heuristic is then used to determine if some weight changes can be applied to reduce the network congestion. We detail below the key steps of the algorithm.

A. Traffic matrix estimation

At time τ, we assume to obtain this data regarding the time interval $\mathcal{I}_\tau$: traffic $\hat{y}_l^\tau$ on each link $l \in E$, together with ingress/egress traffic $b_n^{i,\tau}$ and $b_n^{e,\tau}$ for each edge router n.

To estimate demands at time τ, the "tomogravity" method introduced in [20] is used. The estimated traffic vector $\hat{\mathbf{d}}$ is obtained in a fraction of second, even for large networks.

B. Robust optimization of link weights

A greedy heuristic is used to incrementally solve this problem, iteratively deviating traffic from the most congested links, by increasing their weight. To determine the effectiveness of each deviations, worst-case link loads need to be computed. Thus, this link load evaluation constitute the time consuming key step of the robust optimization heuristic.

We denote by $\mathcal{D}_*^\tau$ the set of traffic vectors $\mathbf{d}$ describing the uncertainty on the traffic at time τ. The following equivalent formulation of problem (METRIC) is obtained:

$$\text{minimize } \rho(\boldsymbol{\omega}) = \max_{l \in E} \max_{\mathbf{d} \in \mathcal{D}_*^\tau} \frac{1}{c_l} y_l(\boldsymbol{\omega}, \mathbf{d})$$

$$\text{subject to } \quad \boldsymbol{\omega} \in \Omega^M$$

The worst-case load of link l can be written as follows:

$$\max_{\mathbf{d} \in \mathcal{D}_*^\tau} y_l(\boldsymbol{\omega}, \mathbf{d}) \quad = \quad \sum_n b_n^{e,\tau} - \min_{\mathbf{d} \in \mathcal{D}_*^\tau} \sum_k (1 - f_{l,k}(\boldsymbol{\omega})) d_k .$$

The computation therefore reduces to solve a minimization problem, whose structure is that of a standard minimum cost flow problem on a bipartite graph, which can be solved very efficiently using a dedicated algorithm [21]. This allows to drastically reduce the worst-case load computing time.

Finally, to reduce the time needed for the traffic to converge on the new routes, the number of weight changes is restricted to the firsts most effective ones.

TABLE I
Topologies characteristics and algorithm execution time

Topology	Nodes	Links	Worst Execution Time (s)
ABOVENET	19	68	2.15
ARPANET	24	100	7.89
BHVAC	19	46	3.35
EON	19	74	3.33
METRO	11	84	1.18
NSF	8	20	0.23
PACBELL	15	42	1.62
VNSL	9	22	0.18

IV. Results

Our results are compared with the static weight configuration where all metrics are set to one, and with the results obtained by the greedy heuristic in the case where the traffic demands are known. We also compare with a lower bound obtained using a multipath routing problem (for which future traffic demands $d_k^{\tau+1}$ are also known at time τ). We emphasize that this lower bound is potentially much lower than the network congestion rate obtained under the optimal weight configuration.

Simulations are performed on 8 real network topologies (see Table I), found in [22] and IEEE literature. For each topology, the initial weight of each link is set to 1. All tests have been done on a Intel Core i5-2430M processor at $2.4\ GHz$, running under Linux with 4GB of memory.

A. Simulated traffic

For each network, a random traffic matrix is generated at time $\tau = 0$. Each minute, the traffic matrix is updated by adding a white gaussian noise, so that each OD traffic demand can vary by at most $\pm 50\%$ over a 5 minutes period.

Figure 2 present the results obtained for the EON topology (similar results are obtained for the other topologies). The x-axis represents the time τ. Two vertical axes are present : the left one is the measured congestion rate of the network. The right one gives the number of weight changes made by the online algorithm. Vertical bars represent the number of link weights modified by the online algorithm. Four curves are plotted on each graph:

- Lowerbound: solution of the multipath problem,
- UnitMetrics: congestion rate obtained with unit metrics,
- KnowMatrixOptim: congestion rate obtained with the greedy heuristic if the traffic matrix was known,
- EstimatedMatrixOptim: congestion rate obtained with the proposed online algorithm.

The adaptive routing scheme allows to significantly reduce the network congestion rate with respect to a static unitary weight configuration. For all considered topologies, there is no significant loss in performance due to the uncertainty on traffic demands. The network congestion rate is often fairly close to the lower bound, indicating that this algorithm provides a near-optimal weight configuration. Finally, the number of metric

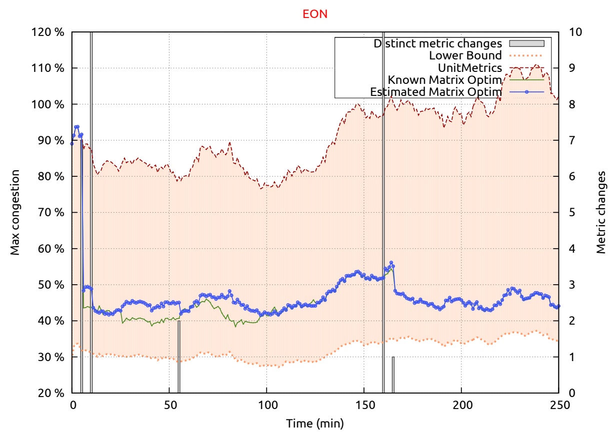

Fig. 2. Congestion for EON topology

changes remains limited, avoiding to continuously change the routes. As showed in table I, the routing decisions are taken in a few seconds, even for large networks, which is clearly compatible with online processing.

B. Real traffic

Similar results are obtained with real public traffic data obtained on the ABILENE network [23]. On the fourth week of the ABILENE recording, compared to unitary metrics, the proposed dynamic routing algorithm allows a significant reduction of the network congestion rate when the traffic is high: the maximum congestion rate of the network is decreased from 55% with unitary metrics to about 35%. The execution time remains also very low: below 0.7 seconds.

REFERENCES

[1] Networking Working Group, "Ospf version 2," Internet Engineering Task Force, Tech. Rep., 1994.

[2] J. Moy, *OSPF, Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.

[3] B. Fortz and M. Thorup, "Increasing internet capacity using local search." *Computational Optimization and Applications*, vol. 29, pp. 13–48, 2004.

[4] C. Fortuny, O. Brun, and J. M. Garcia, "Metric optimization in ip networks," in *19th International Teletraffic Congress*, Beijing, China, 2005, pp. 1225–1234.

[5] B. Fortz and H. Ümit, "Efficient techniques and tools for intra-domain traffic engineering," ULB Computer Science Departement, Tech. Rep. 583, 2007.

[6] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proc. 19th IEEE Conf. on Computer Communications (INFOCOM)*, 2000.

[7] E. Mulyana and U. Killat, "Optimizing ip networks for uncertain demands using outbound traffic constraints," in *INOC'2005*, 2005, pp. 695–701.

[8] A. Altin, P. Belotti, and M. Pinar, "Ospf routing with optimal oblivious performance ratio under polyhedral demand uncertainty," *Optimization and Engineering*, 2009.

[9] A. Altin, B. Fortz, and H. Ümit, "Oblivious ospf routing with weight optimization under polyhedral demand uncertainty," in *International Network Optimization Conference (INOC 2009)*, Pisa, Italy, April 26-29 2009.

[10] J. Chu and C. T. Lea, "Optimal link weights for ip-based networks supporting hose-model vpns," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 778–786, June 2009.

[11] C. Systems, "Cisco ios netflow," http://www.cisco.com/en/us/products/ps6601/products_ios_protocol_group_home.html.

[12] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *ACM SIGMETRICS, San Diego, USA*, June 2003. [Online]. Available: citeseer.ist.psu.edu/article/zhang03fast.html

[13] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffic matrices: balancing measurements, inference and modeling," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 362–373, 2005.

[14] A. Nucci, R. Cruz, N. Taft, and C. Diot, "Design of IGP link weight changes for estimation of traffic matrices," in *IEEE Infocom*, Hong Kong, March 2004.

[15] C. Fortuny, O. Brun, and J. M. Garcia, "Fanout inference from link counts," in *4th European Conference on Universal Multiservice Networks (ECUMN 2007)*, 2007, pp. 190–199.

[16] A. Soule, K. Salamatian, A. Nucci, and N. Taft, "Traffic matrix tracking using kalman filters," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 3, pp. 24–31, 2005.

[17] K. Papagiannaki, N. Taft, and A. Lakhina, "A distributed approach to measure IP traffic matrices," in *Internet Measurement Conference*, 2004, pp. 161–174. [Online]. Available: http://doi.acm.org/10.1145/1028808

[18] T. Ye, H. T. Kaur, S. Kalyanaraman, K. S. Vastola, and S. Yadav, "Dynamic optimization of ospf weights using online simulation," in *IEEE INFOCOM*, 2002.

[19] O. Brun and J. M. Garcia, "Dynamic igp weight optimization in ip networks," in *First International Symposium on Network Cloud Computing and Applications (NCCA)*, 2011.

[20] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *In ACM SIGMETRICS*, 2003, pp. 206–217.

[21] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Prentice Hall, 1993.

[22] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring isp topologies with rocketfuel," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 1, pp. 2 – 16, feb. 2004.

[23] Y. Zhang, "Abilene traffic matrices," http://www.cs.utexas.edu/~yzhang/research/AbileneTM/.