Combined Intra- and Inter-domain Traffic Engineering using Hot-Potato Aware Link Weights Optimization

Simon Balon* and Guy Leduc
Research Unit in Networking
Université de Liège (ULg) - Belgium
{balon, leduc}@run.montefiore.ulg.ac.be

ABSTRACT

A well-known approach to intradomain traffic engineering consists in finding the set of link weights that minimizes a network-wide objective function for a given intradomain traffic matrix. This approach is inadequate because it ignores a potential impact on interdomain routing due to hot-potato routing policies. This may result in changes in the intradomain traffic matrix that have not been anticipated by the link weights optimizer, possibly leading to degraded network performance.

We propose a BGP-aware link weights optimization method that takes these hot-potato effects into account. This method uses the interdomain traffic matrix and other available BGP data, to extend the intradomain topology with external virtual nodes and links, on which all the well-tuned heuristics of a classical link weights optimizer can be applied. Our method can also optimize the traffic on the interdomain peering links.

Categories and Subject Descriptors: C.2.2 [Network Protocols]: Routing Protocols, C.2.3 [Network Operations]: Network management

General Terms: Algorithms, Design, Management, Performance

1. INTRODUCTION & RELATED WORK

Intradomain traffic engineering consists in routing traffic in an optimal way from ingress nodes to egress nodes in a given domain. If shortest path IP routing is used, the only way to optimize the traffic is by finding an appropriate set of link weights that minimizes a given domain-wide objective function. Usually the link weights optimizers (LWOs) consider intradomain traffic matrices to compute the optimal set of link weights. However this approach is unaware of the interdependence between intradomain and interdomain routings. Actually the real traffic demand is an interdomain traffic matrix (from prefix to prefix), while the intradomain traffic matrix (from ingress to egress nodes) is only the result of applying BGP routing decisions on the interdomain traffic matrix (TM). Even if we consider that the interdomain TM and the interdomain (BGP) routes are invariant, the intradomain TM may still vary if some link weights are changed inside the domain. This is due to the so-called hot-potato (or early exit) decision rule implemented by BGP.

A first LWO algorithm for a given intradomain traffic matrix has been proposed by Fortz et al. in [3]. In our LWO we reuse the heuristic detailed in [3], but we have adapted this algorithm to consider the effect of hot-potato routing.

The fact that the intradomain TM is not the correct input for many Traffic Engineering problems had already been pointed out by Feldmann et al. in [2]. In [4] several extensions to the classical LWO problem are briefly described by Rexford, including a sketch of a method that resembles ours. Our work is in line with this recommendation, but our paper proposes a complete method to solve the LWO problem, applicable to intradomain and peering links, and we demonstrate its efficiency on an operational network.

2. A BGP-AWARE LINK WEIGHTS OPTIMIZER

A network is modeled as a directed graph whose vertices and edges represent nodes and links. The basic intradomain topology is composed of all the nodes and links that belong to the AS. For every destination prefix whose egress is chosen with the BGP hot-potato rule we conceptually add a *virtual node* representing it. Then for every peering link on which equivalent BGP routes have been announced for that prefix, we extend the intradomain topology with a link+node pair representing this peering link and the neighboring router on the other side of this link. Finally we attach all these neighboring routers to the virtual node (representing the hot-potato prefix) by adding *virtual links*. To solve the problem in practice we have aggregated all the virtual nodes which had the same set of egress nodes.

Our solution is based on the fact that the traffic will follow the shortest path based on the link weights from an ingress node to the virtual node modelling the destination prefix, which respect both the IGP based on shortest paths and the BGP's hot-potato rule.

3. SIMULATIONS ON AN OPERATIONAL NETWORK

We have tested our algorithm on real data of a multi-gigabit operational network that spreads over the European continent and is composed of about 25 nodes and 40 bidirectional intradomain links. We had access to about one month of traces, one BGP dump per day and one sampled netflow file for each ingress router. With these data we have gen-

*S. Balon is Research Fellow of the Belgian National Fund for the Scientific Research (FNRS) and also partially funded by the EU under the ANA FET project (FP6-IST-27489).

SIGMETRICS'08, June 2–6, 2008, Annapolis, Maryland, USA.
ACM 978-1-60558-005-0/08/06.

erated 2,512 aggregated interdomain traffic matrices (each matrix is an average over 15 minutes). The average number of prefixes is 160,973 of which 97.2% (156,407) are hot-potato prefixes. If we now take traffic into account, we have measured that these 97.2% amount to 35.6% of the traffic on average.

3.1 Intradomain TE

We first compare a classical LWO (denoted *IntraLWO*) with our BGP-aware optimizer (denoted *BGP-awareLWO*). To execute *IntraLWO* we had to generate for each interdomain TM the corresponding intradomain TM where the hot-potato traffic is routed considering the present (i.e., non engineered) link weights. So these intradomain TMs are those that would be measured in the network. For the comparison we have run both optimizers on all the 2,512 aggregated interdomain TMs. Figure 1 shows the maximal intradomain link utilization (U_{max}) for some worst-case TMs.

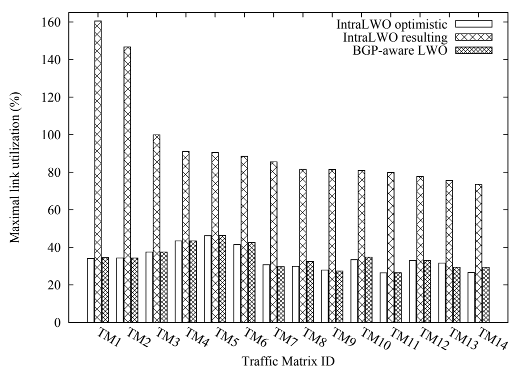

Figure 1: U_{max} values for some worst case TMs

We have run *IntraLWO* on every intradomain TM, and computed the resulting maximal link utilization, assuming that the intradomain TM remains invariant (thus ignoring hot-potato effects). In the sequel these values are denoted *IntraLWO-optimistic*. For this link weights setting, if hot-potato effects are taken into account, we get the resulting maximal intradomain link utilization denoted *IntraLWO-resulting*. These are the real values that would be observed if the optimized link weights were installed in the network. These values are very different, and sometimes the resulting maximal utilization is even worse than the routing without link weight optimization (not present in the figure). Finally we have run our *BGP-awareLWO* and we can see that the maximal link utilizations are very good. Figure 1 shows a selection of TMs providing the worst-case values for *IntraLWO-resulting*. The average reduction of U_{max} from *IntraLWO-resulting* to *BGP-awareLWO* over all TMs is 4.5%, but let us outline that the worst-case TMs do matter much more, because the main goal of our LWO is to filter out the unexpectedly bad link weights settings proposed by a classical LWO. In all cases the real minimal value of U_{max} achievable *in practice* are the values of *BGP-awareLWO*, since the *IntraLWO-optimistic* are disqualified in the comparison.

3.2 Interdomain TE

Now that interdomain links are part of the topology used by the optimizer, we can also balance the load on these links by including them in the objective function. With the present link weights the maximal link utilization of the interdomain links peaks at 73.7%. We have run *BGP-awareLWO* with interdomain links in the objective function and the results are shown in figure 2 for the peak TM. We can see that the maximal interdomain link utilization is reduced from 73.7% to 36.8% when using *BGP-awareLWO*. It shows that the LWO can take advantage of hot-potato routing to also engineer traffic on interdomain links.

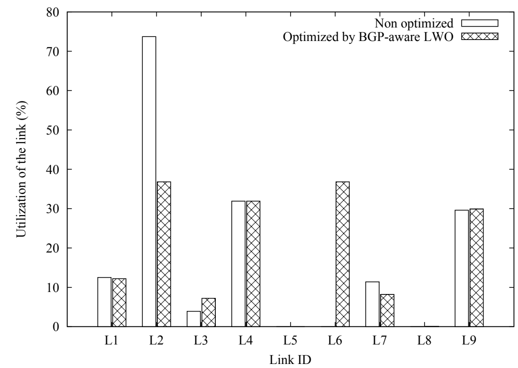

Figure 2: Interdomain link utilizations

4. CONCLUSION

We proposed a BGP-aware Link Weight Optimizer (LWO) that extends the classical (intradomain) LWO to take into account BGP's hot-potato routing principle[1]. The optimized link weights, if deployed, will actually give rise to the link loads expected by the optimizer, contrary to a classical (intradomain) LWO that may lead to unexpectedly high loads on some links when changing weights impacts the intradomain traffic matrix. In practice the method only requires to extend the intradomain topology with a limited number of virtual nodes and links, which preserves scalability, as shown on an operational network used as a case study. The aggregated interdomain traffic matrix associated with this extended topology replaces advantageously the classical intradomain traffic matrix as input to the LWO. On this basis, a classical LWO requires only small modifications to be reused on the extended topology, and this allows us to reuse all its well-tuned heuristics. Our method is also able to optimize traffic on interdomain peering links. We have shown on a case study that it does so very efficiently at almost no extra computational cost.

5. REFERENCES

[1] S. Balon and G. Leduc. Combined Intra- and Inter-domain Traffic Engineering using Hot-Potato Aware Link Weights Optimization. Technical Report arXiv:0803.2824v1[cs.NI], University of Liege - ULg, 2008. http://arxiv.org/abs/0803.2824.

[2] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. NetScope: Traffic engineering for IP networks. *IEEE Network Magazine*, pages 11–19, March/April 2000.

[3] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of INFOCOM*, pages 519–528, 2000.

[4] J. Rexford. *Handbook of Optimization in Telecommunications*, chapter Route optimization in IP networks. Springer Science + Business Media, February 2006.

[1] An extended version of this paper is available at [1].