

Traffic Engineering With Traditional IP Routing Protocols

Bernard Fortz
Institut d'Administration et de Gestion
Universite Catholique de Louvain
Louvain-la-Neuve, Belgium
fortz@poms.ucl.ac.be

Jennifer Rexford Mikkel Thorup
Internet and Networking Systems
AT&T Labs – Research
Florham Park, NJ 07932
{jrex, mthorup}@research.att.com

Abstract

Traffic engineering involves adapting the routing of traffic to the network conditions, with the joint goals of good user performance and efficient use of network resources. In this paper, we describe an approach to intradomain traffic engineering that works within the existing deployed base of Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF) and Intermediate System-Intermediate System (IS-IS). We explain how to adapt the configuration of link weights, based on a network-wide view of the traffic and topology within a domain. In addition, we summarize the results of several studies of techniques for optimizing OSPF/IS-IS weights to the prevailing traffic. The paper argues that traditional shortest-path routing protocols are surprisingly effective for engineering the flow of traffic in large IP networks.

1 Introduction

In some sense, IP networks manage themselves. A host implementing the Transmission Control Protocol (TCP) adjusts its sending rate to the bandwidth available on the path to the destination, and routers react to changes in the network topology by computing new paths. This has made the Internet an extremely robust communication network, even in the face of rapid growth and occasional failures. However, these mechanisms do not ensure that the network runs *efficiently*. For example, a particular link may be congested despite the presence of under-utilized links in other parts of the network. Or, a voice-over-IP call may travel over a route with high propagation delay when a low-latency path is available. Improving user performance and making more efficient use of network resources requires adapting the routing of traffic to the prevailing demands. This task is referred to as *traffic engineering* [1]. In this paper, we focus on engineering the flow of traffic within a single Autonomous System (AS), such as a company, university campus, or Internet Service Provider (ISP).

1.1 Intradomain Traffic Engineering

Traffic engineering depends on having a set of performance objectives that guide the selection of paths, as well as effective mechanisms for the routers to select paths that satisfy these objectives. Most large IP

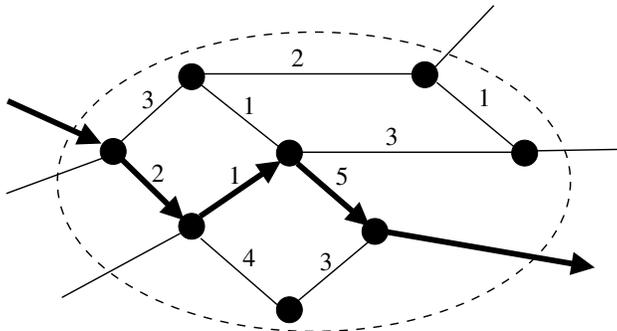


Figure 1: Shortest path routing within an Autonomous System based on OSPF/IS-IS link weights: each link has an integer weight.

networks run Interior Gateway Protocols (IGPs) such as OSPF (Open Shortest Path First) or IS-IS (Intermediate System-Intermediate System) that select paths based on static link weights. These weights are typically configured by the network operators. Routers use these protocols to exchange link weights and construct a complete view of the topology inside the AS, as shown in Figure 1. Then, each router computes shortest paths (where the length of a path is the sum of the weights on the links) and creates a table that controls the forwarding of each IP packet to the next hop in its route.

On the surface, shortest-path routing does not seem flexible enough to support traffic engineering in a network supporting a diverse set of applications. First of all, these IGPs are limited to routing scenarios that can be specified with a single integer weight on each link. However, we argue that link weights suffice to specify near-optimal routing for large, real-world networks. Second of all, in their basic forms, the OSPF and IS-IS protocols do not adapt the link weights in response to changes in traffic or the failures of network elements, and the path-selection process does not directly incorporate any performance objectives. Recent standards activity has proposed traffic-engineering extensions to OSPF and IS-IS to incorporate traffic load in the link-state advertisements and path selection decisions. However, these extensions require modifications to the routers to collect and disseminate the traffic statistics and establish paths based on the load metrics. Instead, we argue that it is often possible to select *static* link weights that are resilient to traffic fluctuations and link failures, allowing the use of the traditional incarnations of OSPF and IS-IS.

The example in Figure 2 shows how to control the distribution of traffic in a network by tuning the IGP weights. All three diagrams concern the same network where all links have the same capacity and each of the nodes q , r , s , and w have one unit of traffic to send to node t . The simple performance objective here is to minimize the maximum link load.

- **Initial configuration with unit weights:** The first diagram shows the results of having the same weight of 1 on every link. This directs all of the traffic from nodes q , r , and s through node u , forcing 3 units of load on link (u, t) .

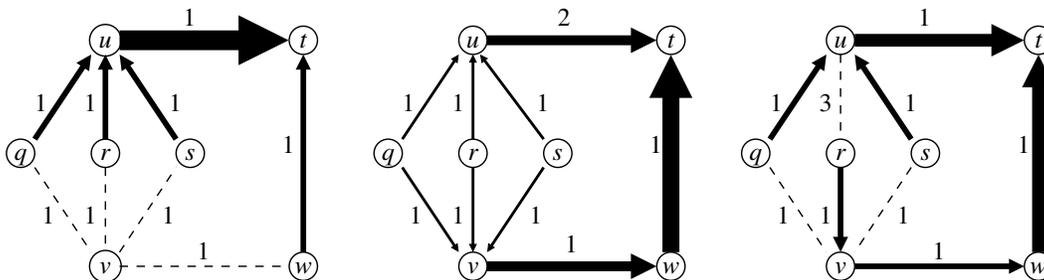


Figure 2: Routing the same demands with different weight settings: each link has an integer weight, arrows illustrate the flow of traffic, the thickness of the arrows indicates the volume of traffic traversing the link, and a dashed line represents a link that carries no traffic.

- Local change to the weight of the congested link:** A naive approach to reducing the load is to increase the IGP weight on the overloaded link (u, t) . In the second diagram, the weight of (u, t) is increased to 2. This configuration results in two shortest paths for nodes $q, r,$ and s , and an even splitting of traffic over paths via u and v . However, this solution places 2.5 units of load on the link (w, t) .
- Global optimization of the link weights:** A global optimization of the weights would produce a weight setting like the one in the third diagram, with no link carrying more than 2 units of traffic. This distribution of traffic is optimal with regard to the maximum load. Since 4 units of traffic have to reach node t along its two incoming edges, no other routing scheme could produce a better solution.

In this example, changing the link weights to alleviate the congestion on the link (u, t) is an attractive alternative to buying and deploying additional bandwidth between routers u and t .

1.2 Advantages of Using Traditional OSPF/IS-IS

This paper presents an overview of a practical approach to working within the existing framework of static link weights, without modification to the routing protocols or the routers themselves. The paper brings together the work in various papers that describe individual components of this approach to traffic engineering. The main point underlying this body of work is that the process of arriving at good values for the weights, or a good set of changes to the existing values of the weights, is handled *externally* from the routers. This process could depend on traffic measurements and topology data collected from the operational network. The selection of the weights may also depend on a wide variety of different cost, performance, and reliability constraints. The link weights are configured by an external entity, such as a network management system or a human operator, to achieve certain traffic engineering goals. Generally, we view a modification of the link weights as a significant change to the network that is performed on a relatively coarse time scale.

The framework we describe has two key features—a centralized approach for setting the routing parameters and the use of link weights as the way to drive the path-selection process. First, setting the routing parameters based on a network-wide view of the topology and traffic, rather than the local views at each router, has the following advantages:

- **Protocol stability:** The routers do not adapt automatically to locally-constructed (potentially out-of-date) views of the traffic. The paths do not change unless the routing parameters are reconfigured or the network topology changes. This predictability aids operators in diagnosing performance problems.
- **Low protocol overhead:** The routers do not need to track changes in load and disseminate new link-state information. This limits the bandwidth consumed by the routing protocol and the computational load imposed on the routers.
- **Diverse performance constraints:** The selection of routing parameters can depend on a variety of performance and reliability constraints. Operators can incorporate constraints that are difficult to formalize in a routing protocol and new constraints (and improved algorithms for selecting the routing parameters) can be incorporated without changing the routing protocols.

Second, using link weights to express the routing configuration has the following advantages:

- **Compatibility with traditional shortest-path IGPs:** Selecting the link weights outside of the network allows operators to engineer the flow of traffic while working with the existing OSPF and IS-IS routing protocols. This avoids the need to upgrade equipment or introduce additional configuration complexity.
- **Concise representation:** Link weights are a concise form of configuration state. Each router is configured with the weight (and perhaps area) for each of its outgoing links. The router does not need any path-level information or any state concerning the incident edges at other routers. In addition, an operator can change multiple paths in the network by changing a single link weight.
- **Default weights and backup routes:** Link weights can have a reasonable default configuration based on link capacity (e.g., inversely proportional to capacity). If the topology changes (e.g., a link failure), the router can automatically compute new routes based on the current topology and link weights. These routes can carry traffic until a new configuration is selected, if necessary.

Some operators already adjust their IGP weights in response to network congestion. In this paper, we show how to tune the IGP weights in a systematic, automated fashion based on measurement data and an optimization algorithm. In Section 2, we outline the key components of a system for assigning link weights based on the traffic demands, network topology, and performance objectives. Next, Section 3 quantifies the effectiveness of shortest-path routing based on link weights in controlling the flow of traffic in the network.

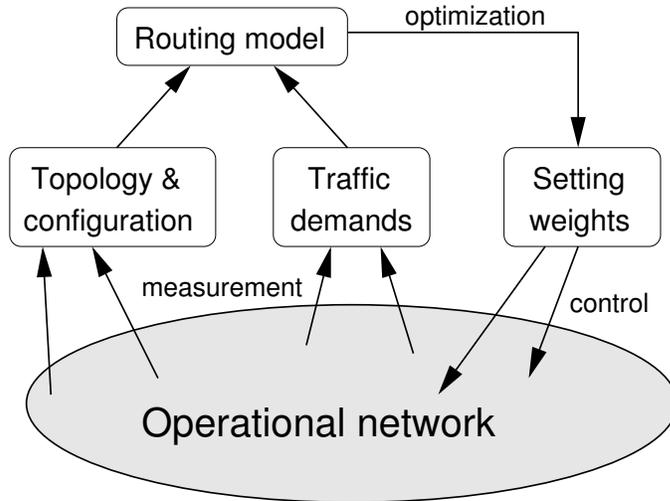


Figure 3: Key components of the traffic engineering framework

We show that good settings of the static link weights allow OSPF and IS-IS perform almost as well as an optimal routing scheme that has complete flexibility in selecting paths for the traffic. Section 4 concludes the paper with a summary of our approach.

2 Traffic Engineering Framework

Our approach to traffic engineering has three main steps—measure, model, and control—as illustrated in Figure 3. First, the operator needs to measure the network topology and the offered traffic; in practice, this measurement data are necessary for other optional tasks, such as network visualization, usage-based billing, and capacity planning. Second, evaluating possible settings of the link weights requires a way to predict how the IGP configuration affects the flow of traffic. Third, after deciding on the values of the weights, an automated system or a human operator needs to change the IGP configuration on one or more routers.

2.1 Measure: Topology/Configuration and Traffic Demands

Selecting good link weights depends on having a timely and accurate view of the current state of the network. This view includes the operational routers and links in the network, as well as the capacity of the links and the current configuration of the IGP parameters (e.g., OSPF/IS-IS weight and area). Topology and configuration information are available from a variety of sources. Link capacity and IGP parameters are available from router configuration data (such as configuration files) and may also be stored in external databases that drive the provisioning of the network elements. The Simple Network Management Protocol (SNMP) provides information about the status of the network elements, either by polling or via traps. In addition, it is possible

to deploy IGP route monitors that track the topology and IGP parameters in the operational network. For example, a software router that participates in OSPF/IS-IS with the operational routers could track and report this information in real time.

The operator also needs an estimate of the volume of traffic between each pair of routers. In some cases, the estimate may come from past experience or customer subscription information. In other cases, a network-wide view of the traffic demands can be gleaned using sophisticated measurement techniques. Four main approaches have been considered [2]. First, the necessary traffic statistics may be available directly from SNMP Management Information Bases (MIBs), depending on the forwarding paradigm employed in the network. For example, Multi-Protocol Label Switching (MPLS) MIBs could be used to measure the volume of traffic on the Label Switched Path (LSP) between each pair of edge routers. Second, the offered traffic can be computed by combining packet-level or flow-level measurements at the network edge with the information available in routing tables. Third, the offered traffic may be inferred based on observations of the aggregate load on links inside the network in conjunction with routing data; this approach is referred to as *network tomography*. Fourth, new techniques for packet sampling offer the possibility of direct observation of the offered traffic as it flows through the network.

2.2 Model: Path Selection Based on IGP Configuration

Traffic engineering requires an effective way to predict the flow of traffic through the network based on the routing configuration. When all of the links belong to a single OSPF/IS-IS area, path selection simply involves computing the shortest path(s) between each pair of routers (e.g., using Dijkstra's algorithm). Larger networks are typically divided into multiple OSPF/IS-IS areas. For routers in different areas, the path selection depends on the summary information conveyed across area boundaries. In some cases, the network may have multiple shortest paths between the same pair of routers. The OSPF and IS-IS protocol specifications do not dictate how routers handle the presence of multiple shortest paths. In practice, most routers capitalize on the multiple paths to balance load. A router typically splits traffic roughly evenly over each of the outgoing links along a shortest path to the destination; traditionally, IGPs with static link weights do not have the flexibility to divide the traffic among the shortest paths in arbitrary proportions.

Ultimately, then, the routing model should compute a *set* of paths for each pair of routers. These paths can be represented in terms of the fraction of the traffic (for this pair of routers) that traverses each of the links. The output of the routing model can be combined with the traffic demands to estimate the volume of traffic on each link, based on the topology and the IGP configuration. The routing model also plays a role in capturing the interaction of the IGP with interdomain routing (i.e., the Border Gateway Protocol (BGP)). A single block of destination IP addresses may be reachable via multiple exit points to neighboring

domains. For example, an AS may have multiple links to another service provider at different geographic locations. The BGP decision process selects from these routes based on the IGP cost of the shortest path to each exit point [3]. This enables each router to select the “closest” exit point. The work in [4] presents an overview of a routing model that captures the details of multiple OSPF/IS-IS areas, the even splitting over multiple shortest paths, and the influence of IGP parameters on how the traffic exits the network en route to a neighboring AS.

2.3 Control: Reconfiguring the IGP Weights

Changing IGP weights requires applying the appropriate commands to the affected routers. This may involve running `telnet` or `ssh` to connect to each router’s command-line interface. The specific commands depends on the particular operating system running on the router. For example, in Cisco IOS (Internet Operating System) parlance, the operator would enter a command like `ip ospf cost 64` in the context of the appropriate outgoing link to change the OSPF weight to 64. These commands may be applied manually by a network operator or performed automatically by a script. More generally, the service provider may have a network management system for configuring the routers. An integrated network management system could conceivably automate the entire process of detecting congestion, selecting suitable IGP weights, and effecting the configuration changes. However, given the complexity of operating a large IP network, a service provider may have a human operator involved to oversee the process.

Following a weight change, the router updates its link-state database and floods the new value of the weight to the rest of the network. Upon receiving the new link-state advertisement, each router updates its link-state database, computes the new shortest paths, and updates certain entries in its forwarding table. During this convergence period, the routers in the network do not have a consistent view of the shortest-path routes for some destinations. This transient period is similar to what happens following a change in the topology of the network due to an equipment failure or the addition of a new router or link; convergence following a weight change is typically faster than convergence after a failure, though, since the router does not have to incur a delay to detect that a failure has occurred. Still, changing a link weight does require the network to undergo a transition period where the forwarding paths are changing for some of the traffic. As such, we do not envision making frequent changes to the link weights. Instead, link weights would change under special circumstances following the outlay of new capacity, a significant equipment failure, or a serious shift in the traffic demands.

3 Performance Properties

In this section, we quantify how well we can engineer the flow of traffic using the traditional OSPF/IS-IS routing protocols in large networks, using an optimization algorithm to identify good IGP weight settings. The algorithm evaluates a number of different weight settings with the goal of satisfying a particular performance objective. In the example in Figure 2, our simple objective was to minimize the maximum link load. When links may have different capacities, it is more natural to consider the link utilization (i.e., the ratio of load to capacity). We generally view a link’s capacity as its maximal desirable load. Our target is therefore to keep the max-utilization below 100%; or, to protect against bursts, our target could be to keep max-utilization below 60%.

The discussion in this section draws on results of [5, 6], to which the reader is referred for technical details and more references to related work.

3.1 Fixed Topology and Traffic Demands

Routing based on link weights is not flexible enough to represent all possible solutions to the routing problem. When judging the quality of a weight setting, we compare it against optimal routing (OPT) that can direct traffic along any paths in any proportions. OPT models an idealized routing scheme that can establish one or more explicit paths between every pair of nodes, and distribute arbitrary amounts of the traffic on each of the paths. Realizing an OPT solution in practice would require the use of a more flexible routing protocol such as MPLS that support explicit routes [7]. It is also important to compare the advanced weight settings with simple default configurations, such as Cisco’s approach of setting the weight of a link inversely proportional to its capacity (InvCapOSPF). Another default is UnitOSPF, which sets each link weight to 1.

3.1.1 Performance Comparison with Max-Utilization

We now return to the example of Figure 2 where nodes q , r , s , and w each send one unit of traffic to t . This time we set the capacity of the “local” links incident to s , r , and t to 1 while the remaining links, (u, t) , (v, w) , and (w, t) , have capacity 2. UnitOSPF gives the weight setting of the first diagram. The maximum utilization is for link (u, v) . With a load of 3 and a capacity of 2 its utilization is 150%. Now, InvCapOSPF will reduce the weights of links (u, t) , (v, w) and (w, t) to $1/2$. This does not affect the routing of traffic from q , r , and s , as it remains shorter to go through u rather than v . Thus, the maximum utilization remains 150%. However, the weight setting in the last diagram achieves a maximum utilization of 100%. The 100% utilization happens on links (u, t) and (w, t) with load and capacity 2, and on links (q, u) , (s, u) , and (r, v) with load and capacity 1. OPT cannot perform any better as the total traffic to t is 4 matching the total

capacity on its two incoming edges.

The fact that OPT cannot outperform the best IGP setting for this particular example does not hold in general. It is easy to construct examples where OPT performs substantially better than the best IGP weights, but these examples often appear quite contrived. The key question is whether a good setting of the weights can balance load as effectively as OPT *in practice*. Several experimental studies have evaluated advanced algorithms that compute good weight settings for a fixed network topology and a given set of traffic demands [5, 8–11]. The exact objective varies from study to study, but they all found good IGP weight settings performing within a few percent of OPT. For example, in [5], for a proposed AT&T IP backbone (with 90 nodes and 274 links) and projected traffic demands, a weight setting was found giving a maximum utilization that was only 3% worse than that of OPT, contradicting the expectation that more flexible routing protocols are necessary for good load balancing. We shall refer to the weight settings from [5] as AdvancedOSPF. In contrast, the simple defaults InvCapOSPF and UnitOSPF performed rather poorly, yielding 50% higher maximum utilization than that of AdvancedOSPF. Similar results were found for other networks.

To appreciate the results, suppose to ensure good network performance, our target is a maximum link utilization below 100%. If our starting point is a default like InvCapOSPF or UnitOSPF, then, switching to AdvancedOSPF, we expect to be able to support 50% higher demands—a very attractive alternative to buying and deploying extra links. Moreover, the AdvancedOSPF weight setting leaves very little room for improvement for any other more flexible routing protocol—no routing protocol can hope to support more than 3% more demand.

3.1.2 Performance Comparison With A Network-Wide Objective

Minimizing the maximum link utilization is a natural and intuitive objective for routing. However, this function is overly sensitive to individual bottleneck links that may be difficult to avoid. For example, an ingress link from a neighboring domain may carry a large amount of traffic under any routing solution. Considering the load on this link to the exclusion of the other traffic in the network does not necessarily result in the best solution. In addition, the maximum link utilization function does not penalize solutions that force traffic to traverse very long paths. For example, if the capacities in Figure 2 were five times bigger, then it would not be worthwhile sending the traffic from r on a detour over v , for it doesn't really matter that we improve the maximum utilization from 30% to 20%.

To deal with these issues, [5] considered an additional network-wide objective. The cost of using a link increases with the utilization, with an explosive growth as the utilization exceeds 100%. Such a link cost function is illustrated in Figure 4. The network-wide cost of a routing solution is then the sum of the link

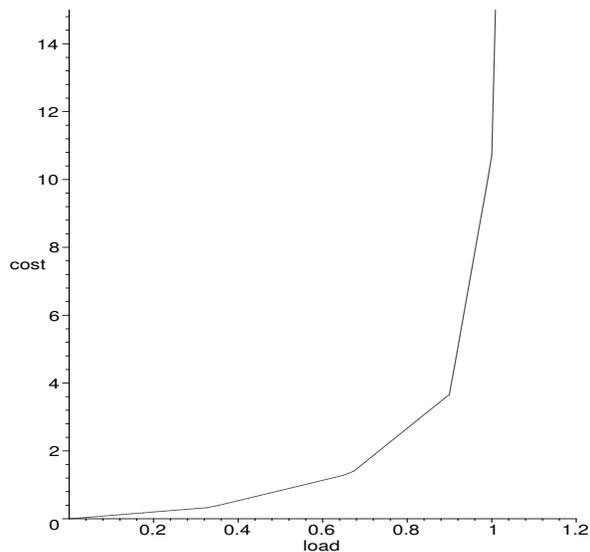


Figure 4: Link cost as a function of the load for a link capacity 1

costs. Figure 5 shows the results for a proposed AT&T backbone with a projected traffic matrix based on traffic measurements and growth trends. Each element in the traffic matrix represents the expected traffic from one router to another. The experiment varied the traffic by multiplying each element by a scaling factor, plotted on the x -axis. The graph plots the network-wide cost normalized to make 1 the threshold for an overloaded network. We see that AdvancedOSPF can handle 70% more demands than the InvCapOSPF and UnitOSPF defaults before passing the threshold. In this graph, OPT is the optimal general routing solution with respect to the network-wide cost, but it can only handle 2% more demands than AdvancedOSPF.

Figure 6 plots the maximum link utilization for same networks, demands, and weight settings as those used for Figure 5, but in Figure 6, OPT is the optimum with respect to maximum link utilization. Thus Figure 6 illustrates how the weight optimization for Figure 5 performs in terms of the maximum link utilization. In particular, we see that the curve for AdvancedOSPF gets very close to OPT when the maximum utilization passes 100%. This is because AdvancedOSPF tries to avoid the high penalty for utilizations above 100%. We see that keeping the maximum utilization below 100%, AdvancedOSPF can handle 50% more traffic than the InvCapOSPF and UnitOSPF defaults whereas OPT can only handle 3% more than AdvancedOSPF, matching our earlier claims for maximum utilization. It is important that the weight settings found by AdvancedOSPF are simultaneously good both with respect to the sum of the link costs and with respect to the maximum link utilizations when it matters. In general, we have found that good weight settings are not very sensitive to the exact details of the objective function. Good weight settings according to one objective function were simultaneously good for other objective functions, as long as the objective function assigns an increasing penalty to links with load approaching capacity.

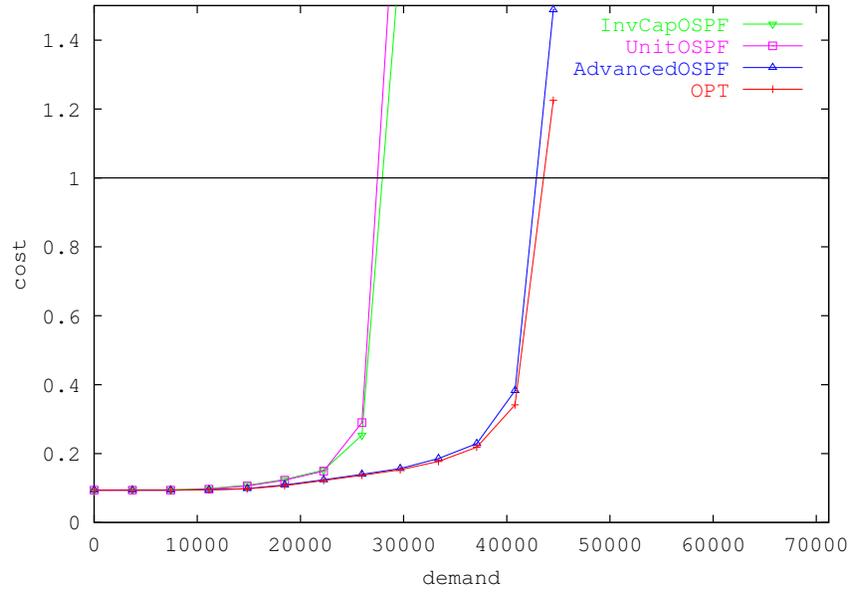


Figure 5: Network-wide cost vs. demand for a proposed AT&T backbone

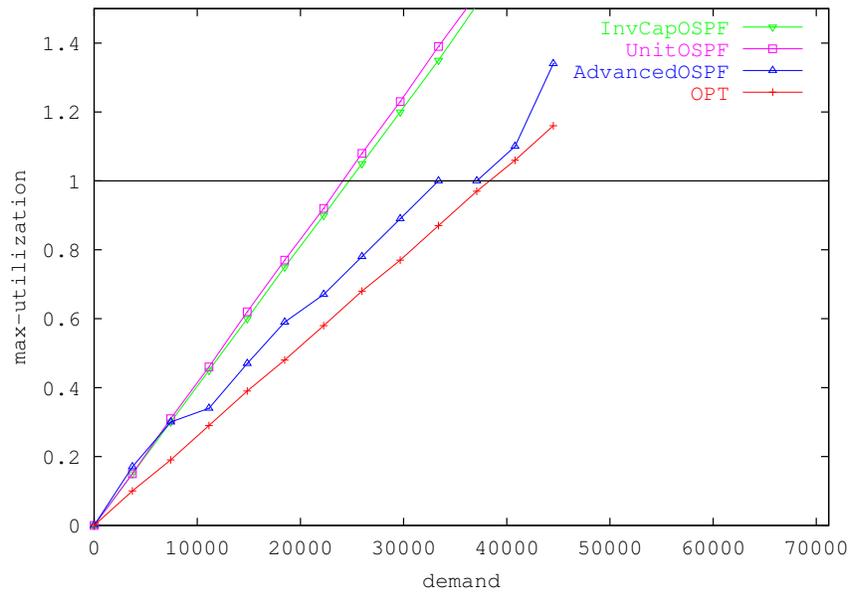


Figure 6: Maximum link utilization vs. demand with same weights as in Figure 5

3.1.3 Reducing the Number of Weight Values

For simplicity, it is appealing to have a small number of different weight values. In addition, having a small set of values significantly reduces the overhead of the algorithm that explores possible changes to the weights. In the experiments, we found that having integer values from 1–20 is sufficient to achieve performance that is competitive with OPT. It was not necessary to use the full range of weight values allowed by the OSPF and IS-IS protocols.

3.2 Changing Traffic Demands

Optimizing the weights for a single topology and traffic matrix is not sufficient. In practice, the traffic volumes fluctuate over time and unexpected failures can result in changes to the network topology. In addition, acquiring an exact estimate of the traffic matrix is difficult. It is important that the setting of the IGP weights be robust to changes in the traffic and the topology. To test robustness, we evaluated the optimized setting of the link weights with different traffic matrices. Traffic fluctuations and errors in measuring the traffic matrix were captured by introducing noise. Each element of the traffic matrix was multiplied by a random number between 0 and 2. Although the demands do not change in expectation, this changes each element by 50% on the average. Still, weight settings based on the original traffic matrix continued to perform well for the new input. An optimized setting of the link weights makes good use of the capacity between various parts of the network, and this is not sensitive to small or moderate changes in the traffic.

In addition to statistical fluctuations, the traffic matrix changes throughout the day. For example, example, there may be substantial structural differences between day and evening traffic. We found that IGP weights optimized for the day-time traffic do not necessarily perform well for the night-time traffic. Instead, we can select a single weight setting that accounts for two different traffic matrices. For example, the optimization can minimize the maximum link utilization across *both* traffic matrices. In our experiments, we found that a single weight setting could perform quite well for both traffic matrices—almost as well as weights optimized for each matrix individually. Using this single weight setting for the entire day has two main advantages. First, operators do not need to disrupt the network with changes to any of the link weights. Second, the common weight setting performs well for all convex combinations of the two demand matrices and, as such, is effective at accommodating the gradual transitions between day and evening traffic. The approach of optimizing for multiple demand matrices can also be used to select IGP weights that satisfy the requirements of multiple classes of traffic in a network that supports differentiated services.

3.3 Few Changes to the Link Weights

Ultimately, changes to the link weights are necessary in response to large shifts in the traffic and certain router or link failures (particularly situations that cause multiple links to fail at the same time). Limiting the number of weight changes is important to limit the disruption to the network. Fortunately, changing a single link weight is often quite effective. When evaluated for an operational AT&T topology, we found that increasing a single weight from 1024 to 1025 could reduce the maximum link utilization by 8%. We also evaluated the effects of all possible single-link failures to see how often a change in the topology would require a new setting of the weights. In almost every case, the existing IGP weights continued to perform relatively well even after a single link failure. The maximum link utilization remained close to the minimum value. However, the failure of a few critical links would require changes to the link weights. A single weight change is often sufficient to alleviate congestion that would arise after a link failure; allowing up to three weight changes was enough to return within a few percent of routing optimized to the new topology. As a proactive measure, the necessary weight changes could be computed in advance of any link failure and stored by the network management system.

Finding good routing configurations with a small number of weight changes suggests that operators could follow a hybrid approach in assigning the link weights. Setting the weights inversely proportional to link capacity has the advantage of being intuitive and simple. A human operator inspecting the router configuration could easily “eye-ball” the weight settings to identify any unusual patterns. Starting with weights inversely proportional to link capacity, we found that changing the weights for just ten of the links was sufficient for the network to perform very close to OPT. This has two important implications. First, operators could still “eye-ball” the configuration, while noting that a few unusual links may have a different weight setting. Second, the algorithm that searches for good weight settings can use the inverse-capacity configuration as a starting point. This reduces the number of scenarios that the algorithm must explore before finding a suitable configuration of the IGP weights.

4 Conclusions

Intradomain routing protocols such as OSPF and IS-IS have been deployed in a large number of networks throughout the Internet for many years. In this paper, we have described an approach to engineering the flow of traffic in these networks by monitoring the traffic and topology, optimizing the setting of the static link weights, and reconfiguring the routers with new weight settings as needed. This approach treats traffic engineering as a network operations task, rather than the responsibility of the underlying routing protocol. Working with the traditional incarnations of OSPF and IS-IS has many practical advantages over using the

proposed traffic engineering extensions to these protocols. Experimental studies show that shortest-path routing protocols can be made to perform quite well. The approach described in this paper can be applied today in a variety of operational networks, and can help guide the decisions of how to instrument these networks to collect accurate information about the topology and the traffic demands.

The ultimate decision of whether to deploy and use more flexible routing protocols (such as the traffic-engineering extensions to OSPF and IS-IS) may depend on factors beyond the load-balancing issues discussed in this paper. For example, more advanced routing protocols can support route “pinning,” which allows the movement of some traffic from one path to another without disrupting the paths for other traffic. In addition, some protocols support the establishment of backup paths, allowing faster re-routing in the event of a network failure. These two enhancements have the potential to reduce the temporary disruption of existing traffic in the face of routing changes and network failures. The features may be useful for networks that support real-time applications or have relatively frequent equipment failures. However, the performance benefits and operational complexity of these enhancements need to be better understood. The results in Section 3 can help inform the discussion of the cost-performance trade-offs. Despite the advantages of the more advanced routing protocols, we believe that the simple approach of adjusting the static link weights may remain viable for many IP networks.

Acknowledgments

We would like to thank Randy Bush for his feedback on an early outline of this paper, and Aman Shaikh for his comments on our initial draft. We are also grateful to Jay Borkenhagen from AT&T for his help in understanding the operational constraints for configuring OSPF in large IP networks. We would also like to thank the anonymous referees for their suggestions on ways to improve our paper.

References

- [1] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, “Overview and principles of Internet traffic engineering.” Request for Comments 3272, May 2002.
- [2] M. Grossglauser and J. Rexford, “Passive traffic measurement for IP operations,” in *The Internet as a Large-Scale Complex System*, Oxford University Press. To appear, 2002. <http://www.research.att.com/~jrex/papers/sfi.ps>.
- [3] S. Halabi and D. McPherson, *Internet Routing Architectures*. Cisco Press, second ed., 2001.
- [4] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, “NetScope: Traffic engineering for IP networks,” *IEEE Network Magazine*, pp. 11–19, March 2000.
- [5] B. Fortz and M. Thorup, “Internet traffic engineering by optimizing OSPF weights,” in *Proc. IEEE INFOCOM*, March 2000.

- [6] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 756–767, May 2002.
- [7] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communication Magazine*, pp. 42–47, December 1999.
- [8] F. Lin and J. Wang, "Minimax open shortest path first routing algorithms in networks supporting the SMDS services," in *Proc. International Conference on Communications*, vol. 2, pp. 666–670, 1993.
- [9] K. Ramakrishnan and M. A. Rodrigues, "Optimal routing in shortest-path data networks," *Bell Labs Technical Journal*, vol. 6, January–June 2001.
- [10] L. S. Buriol, M. G. C. Resende, C. C. Ribeiro, and M. Thorup, "A memetic algorithm for OSPF routing," in *Proceedings of the 6th INFORMS Telecom*, pp. 187–188, 2002.
- [11] M. Ericsson, M. Resende, and P. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," *J. Combinatorial Optimization*, vol. 6, no. 3, pp. 299–333, 2002.