Traffic Engineering Between Neighboring Domains

Jared Winick, Sugih Jamin
Electrical Engineering & Computer Science
University of Michigan
Ann Arbor, MI 48105
{jwinick,jamin}@eecs.umich.edu

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

Abstract— **Network operators control the flow of traffic in today's IP networks by manipulating the configuration of the routing protocols running on their routers. Although intradomain traffic engineering is largely isolated to individual Autonomous Systems (ASes), changes to interdomain routing policies affect the traffic load in other domains. In this paper, we argue that neighboring ASes should cooperate when changing how traffic flows between domains. Rather than proposing a new routing or signaling protocol, we argue that neighboring ASes can work in the context of the existing interdomain routing protocol—BGP (Border Gateway Protocol)—by employing network management tools that predict how configuration changes would affect the flow of traffic. Before modifying the configuration of the live routers, network operators can evaluate the effects of a proposed change and convey this information to the neighboring domain. This domain can, in turn, apply the same kind of tool to evaluate the effects of the proposed configuration change. We describe how to limit the amount of information that the neighboring ASes must exchange. This is crucial for the scalability of our scheme and for protecting the proprietary data of each domain. We also describe how a pair of ASes can control the traffic traveling between their networks without affecting how traffic flows through other domains not involved in the negotiation.**

I. Introduction

Traffic engineering involves tuning a network's resource allocation policies to the prevailing traffic. For example, suppose that measurement data (such as SNMP statistics) show that a link inside the network is overloaded. Diverting some of the traffic to other paths could alleviate the congestion and improve the performance experienced by users. A network operator could change the paths by modifying the configuration of the intradomain routing protocol running on one or more of the routers. For example, Interior Gateway Protocols (IGPs) like OSPF and IS-IS select shortest-path routes based on the sum of integer link weights. Changing the link weights triggers the selection of new paths for some portion of the traffic. For the most part, the effects of IGP configuration changes are local to the operator's network. However, traffic engineering grows more complicated if the operators need to alter how packets travel *between* domains, since these changes have direct effects on the flow of traffic in other networks.

Most of the traffic carried by a large IP backbone traverses multiple Autonomous Systems (ASes). An operator may need to change how traffic flows to or from neighboring ASes for a variety for reasons. First, the two ASes may be installing a new link or upgrading the capacity of an existing link. Exploiting the additional bandwidth typically requires making changes in how packets travel between the two domains. Second, an existing edge link may be overloaded, requiring the operator to divert some of this traffic to a different link to the neighboring AS. Third, the pair of ASes may have a peering agreement that limits the load in each direction on the edge links, which may require an operator to divert some traffic to a different next-hop AS. Fourth, an operator who observes (say, through active measurement) that customers are experiencing poor end-to-end performance may direct the traffic to a different edge link along a path with a higher bottleneck throughput. In each of these examples, a routing change introduced in one AS has an effect on the flow of traffic in another network. The traffic change could cause unforeseen problems, such as overloading of a link inside the neighboring AS or changing how the traffic travels from the neighbor to other parts of the Internet.

The state-of-the-art for interdomain traffic engineering is extremely primitive. Interdomain routing depends on the Border Gateway Protocol (BGP) [1], a path-vector protocol that does not incorporate any

performance, load, or capacity information. BGP operates at the level of address blocks, or prefixes; for example, $192.0.2.0/24$ represents the 256 addresses ranging from $192.0.2.0$ to $192.0.2.255$. A router sends an announcement to notify its neighbor of a new route to the destination prefix and sends a withdrawal to revoke the route when it is no longer available. Each advertisement includes a number of attributes about the route, including the list of ASes along the path to the destination prefix. The router applies import policies to filter unwanted routes and to manipulate the attributes of the remaining routes. Ultimately, the router invokes a decision process to select exactly one "best" route for each destination prefix. The router then applies export policies to manipulate attributes and decide whether to advertise the route to neighboring ASes. Router vendors provide a wide variety of configuration commands for composing the import and export policies.

In today's Internet, operators have *indirect* control over the flow of traffic by configuring import policies that favor some routes over others. BGP offers some basic mechanisms for neighboring ASes to influence each other's routing choices (e.g., by setting the community and multiple exit discriminator attributes in the BGP advertisements). However, the control is primitive due to the complex nature of the BGP decision process and the lack of information about link (and path) capacity and traffic load. Controlling how traffic *enters* a network is especially difficult, and typically relies on ad hoc methods such as "AS prepending" that artificially inflate the length of an AS path; operators typically cannot predict how these techniques affect the flow of traffic. Interdomain traffic engineering in today's Internet is an art where operators have to rely on "tweak and pray". This is increasingly becoming an unacceptable way to engineer large IP networks, as users have begun to expect more predictable communication performance.

On the surface, a natural approach to this problem would be to develop new "Quality of Service" routing protocols that govern the selection of AS-level (or router-level) paths through the Internet. The routers could exchange information about the traffic load and resource constraints to compute paths that satisfy the end-to-end requirements of each "flow". However, scaling QoS routing (and the associated signaling) to a network the size of the Internet would be extremely difficult, if not impossible. In addition, interdomain routing in the Internet depends on complex local policies that relate to the commercial relationships between the ASes. Different ASes may have very different goals, constraints, and policies. As such, it would be extremely difficult for a single set of QoS metrics to drive the selection of end-to-end paths. For these reasons, we do not advocate developing an interdomain QoS routing protocol. Instead, we propose treating interdomain traffic engineering as a network management task that is supported by appropriate network management tools and relationships between the affected parties.

This paper proposes that neighboring ASes cooperate to control the flow of traffic in the context of the existing BGP protocol. Still, we believe it is important for ASes to *coordinate* before making significant changes in how traffic flows between them. An AS should not "act alone." We argue that interdomain traffic engineering should involve three main steps: (i) exploring local changes in BGP policies and IGP weights; (ii) conveying the influence of these changes to the affected neighboring domain(s); and (iii) allowing the neighboring domains to evaluate the impact on their networks before "accepting" the change. This process could repeat several times before settling on a mutually agreeable change in the configuration. For example, an AS may want to divert outbound traffic from one egress link to another. Yet, the neighboring domain may realize that this change would overload a key link inside its network. An alternative configuration change might have a better effect on the neighbor and result in better end-to-end performance. Our proposed approach introduces two main challenges—how to predict the influence of local configuration changes on the flow of traffic and how to conduct the negotiation between the neighboring ASes. The first issue has been addressed in previous work [2, 3]. The second issue is the focus of this paper.

In this paper, we first argue that an AS should limit the amount of information it conveys to its neighbor. This is important for the scalability of our approach and also to protect proprietary information about the traffic, topology, and routing policies in each domain. We show how to minimize the information exchange while still allowing the neighboring AS to evaluate the influence of the configuration changes. Then, we argue that neighboring ASes should help each other select configuration changes that do not affect how traffic enters or leaves other domains not involved in the negotiation. We show how to select configuration changes that scope the effects on

other ASes in this manner. Together, these two techniques provide an effective way for neighboring ASes to control the flow of traffic. In addition to limiting the influence on other ASes, our techniques are incrementally deployable—any pair of neighboring ASes (such as two large service providers, or a customer and its provider) could employ these techniques without the support of the rest of the Internet. We present a preliminary analysis of traffic and routing data from AT&T's commercial IP backbone to demonstrate that our proposed techniques are viable.

II. Traffic Engineering Framework

In this section, we argue that a pair of neighboring ASes can work together to control the flow of traffic between them. Each AS can apply a network management tool that predicts how BGP policies and IGP parameters affect the flow of traffic. An AS that wants to change its routing configuration can provide the information that the neighbor needs to predict the effects on its network. We describe how to aggregate this traffic data to improve scalability and hide sensitive information. Then, we discuss how an AS can avoid configuration changes that would have side effects on how traffic flows through other ASes that connect to the neighboring domain.

A. Predicting the Traffic Demands

Operators can use network management tools to predict the influence of changes in BGP policies and IGP parameters *before* altering the configuration of the live routers [2, 4]. Such tools rely on instrumentation of the operational network to provide an up-to-date snapshot of the network topology, the BGP routes advertised by neighboring domains, and the configuration of the local routing policies/parameters, as well as the offered traffic, as shown in Figure 1. Operators can apply these tools to perform "what if" analysis of potential changes to the routing configuration. In practice, the operators may limit these changes to small, incremental modifications of the existing configuration. These changes may be chosen with certain goals in mind, such as limiting the configuration and protocol overhead, ensuring that the changes have a predictable influence on the flow of traffic, and avoiding sensitivity to small changes in the BGP routes advertised by neighboring domains. The work in [3] describes effective ways to modify BGP policies to engineer the flow of traffic while adhering to these goals.

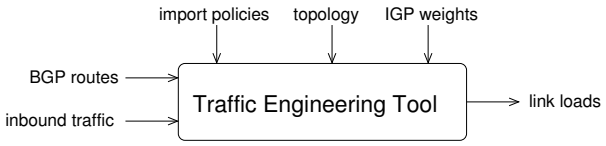

Fig. 1. Predicting the effects of changes to routing configuration

In this paper, we are concerned with the fact that configuration changes made in one domain have an effect on the outbound traffic that flows to one or more neighboring ASes. We propose that, before changing the configuration of the live routers, the operators in the neighboring domain(s) should be given an opportunity to evaluate the effects on the flow of traffic in their network(s). In particular, a neighboring domain could apply a similar kind of prediction tool to evaluate how the change in where traffic enters the network would influence the flow of traffic. This would allow the operators in the neighboring domain to provide feedback about the proposed change or prepare for configuration changes of their own to accommodate the new distribution of traffic. In this paper, we investigate what kind of information the neighboring domains need to exchange to drive this exploration of possible changes in routing configuration. In particular, we are concerned about controlling the amount of information and limiting the effects on other ASes not involved in the negotiation. We defer the discussion of the details of a specific protocol for exchanging this information to future work.

Consider an AS A that is making a configuration change that will affect the flow of traffic to AS B. A simple approach would have AS A inform AS B of the volume of traffic destined to each destination prefix via each of the edge links between the two domains. For example, AS A could indicate that 50 Mbits/second of traffic destined to 192.0.2.0/24 would enter AS B through link 1 and 30 Mbits/second would enter via link 2. Upon receiving this information, AS B could identify the places where this traffic would leave its network en route to the final destination. For example, suppose that AS B sends traffic destined to 192.0.2.0/24 via one of two edge links 5 and 6 to another AS C. AS B could use its network management tool to determine that the traffic entering via 1 might exit on link 5 and the traffic entering via 2 might exit via link 6. The tool could also determine how this traffic would flow through AS B's network and compute the resulting

load on each of the links. Based on these results, AS B might determine that this new distribution of traffic is acceptable and convey a positive response to AS A. Alternatively, the configuration change may overload some link in AS B's network, causing AS B to request that AS A consider a different configuration change.

B. Aggregating the Traffic Data

The strawman approach described in the previous subsection requires AS A to provide a large amount of information to AS B. A typical default-free Internet routing table contains routes for more than 120,000 destination prefixes. Sending detailed traffic statistics to AS B for a large number of these prefixes could introduce significant overhead. In addition, AS A may not want to divulge such fine-grain measurement data. In fact, such detailed traffic statistics might obviate the need for AS B to collect such detailed measurements of its own. Instead, AS A should limit the amount of data to minimize overhead and hide sensitive information from AS B. AS A should provide just enough information for AS B to be able to predict the effects of the proposed configuration change, based on AS B's own measurement data. Previous work introduced the notion of a "traffic demand"—a volume of traffic at a single ingress point that travels to a destination prefix reachable via a particular set of egress points [5]. For example, the volume of traffic entering at link 1 that is destined to a prefix reachable via links 5 and 6 would be a single traffic demand in AS B's network.

Rather than providing traffic statistics on a per-prefix basis, we argue that AS A can send aggregate information about the prefixes that would be routed the same way in AS B's network (i.e., the "traffic demands" in AS B's network). In particular, AS A can aggregate traffic destined to prefixes that have the same set of egress points in AS B's network. AS B can help guide this aggregation by identifying a list of prefixes that can be grouped together—note, though, that AS B does *not* need to identify which of its egress links are used to direct traffic toward these destinations prefixes. Previous studies [3, 6, 7] have shown the potential for grouping related prefixes that have the same Internet routes[1]. In this paper, we argue that this grouping can improve the scalability of the information exchange between the neighboring ASes and can also hide the exact details of the volume of traffic destined to each individual prefix. In addition, the grouping of prefixes could also simplify the collection of the measurement data by allowing AS A to measure traffic volumes for a relatively small number of groups of prefixes in lieu of computing fine-grain per-prefix statistics.

To further reduce the overhead of collecting and sending the data, the two ASes could agree to focus on a small number of destination prefixes (or a small number of groups of related prefixes). Numerous studies of IP traffic measurements have shown that a small fraction of the destinations receive the bulk of the traffic. As such, narrowing the attention to a small set of popular destinations would still allow the two ASes to manipulate a large volume of traffic. To reduce the amount of information exchange, AS A could inform AS B only of the *changes* in the outbound traffic—for the groups of prefixes that have traffic moving from one ingress point to another. Assuming that AS A proposes relatively small, incremental modifications to the existing configuration, this should offer a substantial reduction over sending information about all of the popular groups of destinations. Finally, AS A does not need to convey the absolute traffic volumes for the traffic that moves from one edge link to another. Instead, AS A could provide relative statistics (e.g., "70% of the traffic traversing link 2 destined to group 7 is moving to link 3"). This requires AS B to collect sufficiently detailed traffic statistics of its own to know how to translate the "70%" to an absolute number.

C. Limiting Impact on Other ASes

A key goal of our approach to interdomain traffic engineering is to ensure that changes in the flow of traffic only affect the ASes involved in the negotiation. The work in [3] considers how AS A can make changes without affecting its own downstream customers; in this paper, we consider how AS A can avoid making routing changes that affect the neighbors of AS B and other domains in the rest of the Internet. When AS A and AS B change the flow of traffic, no other ASes should see a change in how traffic flows through their networks. This ensures that local changes do not propagate through the global Internet. This allows our techniques for interdomain traffic engineering to deployed incrementally.

Referring to the example in Figure 2, suppose that link 2 is heavily loaded with traffic flowing from AS A to AS B. AS A may consider moving traffic des-

[1] AS B would need to inform AS A of changes to the grouping of prefixes over time. Preliminary analysis of BGP routing updates suggests that these groupings are relatively stable.

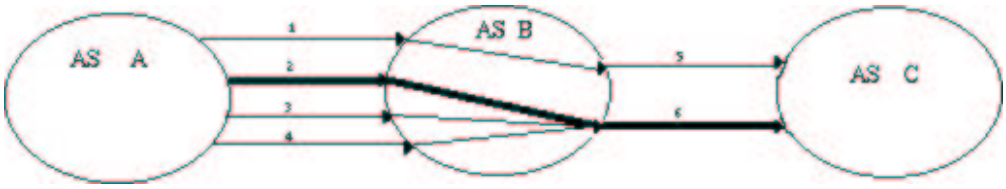

Fig. 2. AS A sending traffic via AS B to destinations in AS C

tined to a prefix in AS C from link 2 to a different egress link to AS B. Suppose the traffic destined to prefix d that leaves AS A via link 2 would leave AS B via link 6, as indicated by the bold path in Figure 2. This imposes a constraint on where AS A can move this traffic without affecting AS C. In particular, AS A cannot move the traffic to link 1, because this would change where the traffic leaves AS B; AS C would experience an increase in traffic on link 5. If the final destination d of the traffic is several AS hops away from AS B, moving the traffic from link 2 to link 1 may change the flow of traffic in other parts of the Internet as well. Instead, AS A should only move the traffic to an edge link that does not change how the traffic would exit AS B. For example, AS A could safely move traffic destined to d from link 2 to links 3 and 4, with no effect on AS C. In general, AS A is limited to moving traffic to other links that ultimately lead to the same egress point in AS B.

In this example, links 2, 3, and 4 are "equivalent" with regard to the destination prefix d, in the sense that AS A can move traffic from one link to another without affecting how the traffic leaves AS B. Clearly AS A cannot be aware of the equivalence of links 2, 3, and 4, on its own. AS B must inform AS A about which links are "equivalent" for a given destination prefix. This forms a partition of the egress links in AS A that are associated with the destination prefix. In this example, AS A has an egress set with links 1, 2, 3, and 4 for destination d, with a partition $E_d = \{(1), (2, 3, 4)\}$. Rather than AS B needing to tell AS A the partition of the egress set for every destination prefix which it announces, destinations which share the same partition can be aggregated.

Restricting AS A to moving traffic among "equivalent" egress links imposes a limitations on interdomain traffic engineering. To quantify these limitations, we analyzed traffic and routing data from the AT&T commercial IP backbone. We treated AT&T as AS B in Figure 2 and used BGP table dumps and Cisco Netflow data [8] to generate a view of where traffic entering on peering links leaves the network en route to the customers. A BGP routing table entry

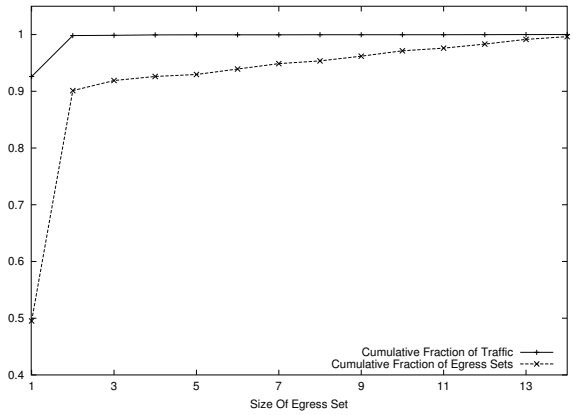

Fig. 3. Cumulative distributions of the fraction of egress sets and traffic per egress set size

consists of a destination prefix, next hop, as well as several other attributes. Examining the BGP tables at the peering routers shows where incoming traffic destined to a particular prefix will leave (via a particular "next hop" that corresponds to a router connecting to customers). At each peering router, we aggregated the Netflow data to compute the total volume of ingress traffic for each destination prefix. Netflow measurements were collected over the 24 hours of April 1, 2002, and the BGP routing tables were dumped at approximately 2 a.m. on the same day.

Using the next hop IP addresses across the BGP tables, we computed the egress set for each destination prefix. The size of the egress set in AS B is useful for describing the flexibility AS A has in moving traffic to alternate edge links between the two ASes. A small egress set in AS B's network tends to imply that AS A has a great deal of flexibility in where traffic can enter AS B while still maintaining the same egress point in AS B. When the traffic has a single egress point in AS B, then AS A can direct the traffic to *any* edge link without affecting how the traffic leaves AS B's network. In contrast, a large egress set in AS B implies that the selection of a specific egress point in AS B may be strongly dependent on where traffic enters the network from A.

The bottom curve in Figure 3 plots the cumulative

distribution of the fraction of egress sets that have a certain number of egress points. Nearly 50% of the egress sets consist of a single egress router. The top curve plots the fraction of inbound traffic associated with these egress sets. The destination prefixes that have a single egress point account for almost 94% of the traffic. Upon further inspection, we find that many of these destinations are associated with university campuses or broadband providers with prefixes that are reachable at a single access point. The Netflow data show that these prefixes receive a large amount of peer-to-peer (P2P) traffic.

The large amount of traffic with a single egress point suggests that AT&T's peers would have substantial flexibility in moving traffic from one edge link to another without affecting how traffic flows to AT&T's customers. To verify this conclusion, we analyzed the inbound traffic from five major ISPs that connect to AT&T at various locations throughout the network. This analysis treats each of these five ISPs as AS A in Figure 2 and quantifies the degree of flexibility in moving traffic between the edge links connecting to AT&T. For each peer, more than 90% of the traffic could be freely moved among *all* of the edge links without affecting the egress point in AT&T's network. Less than 0.1% of the incoming traffic could not be moved to *any* other ingress point without affecting how traffic would leave the network. These results, which held consistently across all five of the peers, suggest the viability of coordinating the flow of traffic between pairs of ASes without necessarily involving the rest of the Internet.

III. Conclusion

This paper proposes an approach to interdomain traffic engineering based on a loose collaboration between individual pairs of ASes. The two ASes can work together to find a mutually agreeable way to engineer the flow of traffic without divulging sensitive information to each other and without affecting how the traffic travels to other domains. The approach is incrementally deployable and works with the existing suite of routing protocols. A complete solution would need to address several key issues:

- *Control protocol:* The neighboring ASes need to have a protocol that defines how they interact—how they identify groups of related prefixes (for aggregation), groups of related edge links (that can be freely used to move traffic from one point to another), changes in traffic volumes, and acceptance/rejection of proposed modifications to the flow of traffic.
- *Selecting a good solution:* The neighboring ASes need to explore one or more possible changes to the flow of traffic. The number of candidate configuration changes may be quite large. Effective ways to narrow the search space, perhaps based on hints exchanged between the two domains, would be extremely useful. Also, the iteration between the two domains should not encourage either AS to "game" the system by providing misinformation.
- *Inbound traffic:* An AS has relatively limited control over how traffic enters the network. Ideally, an AS should be able to instruct its neighbor(s) to alter how they send traffic out their various egress links. In this case, AS A would change its routing configuration at the request of AS B. Finding effective ways for AS B to communicate its traffic engineering requirements to AS A would be quite useful.
- *Traffic measurement:* Traffic measurement plays a crucial role in predicting the influence of configuration changes and in providing information to the neighboring domain. Effective ways to collect relatively fine-grain traffic measurement data would be an important part of our proposed scheme. Flexible ways to *adapt* the collection of the data as needed (say, to measure certain groups of related prefixes) would be very valuable.

We are investigating these issues as part of our ongoing work.

References

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol." Request for Comments 1771, March 1995.

[2] N. Feamster and J. Rexford, "Network-wide BGP route prediction for traffic engineering," in *Proc. Workshop on Scalability and Traffic Control in IP Networks, SPIE ITCOM Conference*, August 2002.

[3] N. Feamster, J. Borkenhagen, and J. Rexford, "Controlling the impact of BGP policy changes on IP traffic," May 2002. `http://www.research.att.com/~jrex/papers/bgpte.ps`.

[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] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. Networking*, vol. 9, June 2001.

[6] A. Broido and K. Claffy, "Analysis of RouteViews BGP data: Policy atoms," in *Workshop on Network-Related Data Management*, May 2001.

[7] T. Bu, L. Gao, and D. Towsley, "On routing table growth." `ftp://gaia.cs.umass.edu/pub/Bu_routingtable_01.ps.gz`.

[8] "Cisco Netflow." `http://www.cisco.com/warp/public/732/netflow/index.html`.