Towards Coordinated Interdomain Traffic Engineering

Ratul Mahajan† David Wetherall† Thomas Anderson†

Abstract – We argue that today's ad hoc inter-domain traffic engineering techniques be replaced with an architecture that is based on explicit coordination between ISPs. With explicit coordination, ISPs exchange information about their traffic and routing options, and all ISPs impacted by a potential routing change negotiate the actual change. This increases efficiency because it allows ISPs to find "win-win" routing outcomes that benefit both relative to routing without coordination; one ISP need not guess the policies of others to successfully engineer its own network. It also offers greater stability because it makes resource policy conflicts visible before routing changes uncover them, allowing them to be resolved and avoiding inadvertent violations. We sketch the building blocks of such an architecture, and use simulation to show it has the potential to yield significant cost benefits.

1. INTRODUCTION

The original design of BGP provided for reachability across individual ISP networks [17] but did not support routing policies based on performance, cost, load, or other dynamic metrics, perhaps because of stability concerns [7]. In hindsight, this has proven to be a serious shortcoming. ISPs need to implement such policies whether the protocol includes them or not: networks do become overloaded, e.g., when their operating capacity is reduced by failures or exceeded by flash crowds, and routing options do significantly affect cost and performance.

To meet these needs, existing BGP mechanisms have been co-opted over time, and new ones have been invented to solve specific problems. Outgoing traffic can be controlled using local-prefs and "smart routing" [27, 15]. Incoming traffic can be influenced using AS-path prepending, prefix splitting, selective announcements, MEDs and communities [24]. Such mechanisms are gaining widespread adoption. For instance, half of the unique AS-paths in the current BGP tables are prepended [22], and smart routing is becoming available in the form of commercial products.

Unfortunately, this ad hoc collection of techniques suffers from systemic problems that will only worsen with increased usage. A fundamental issue is that the current mechanisms are designed to be used unilaterally, i.e., at the discretion of one ISP, even though they directly impact other ISPs. This can lead to serious instabilities because ISPs can inadvertently violate each other's resource policies. In one recent example, two large ISPs were involved in an incident that lasted for two days [14]. Each ISP was making independent changes to engineer its own network. But, inadvertently, these changes adversely impacted the other ISP which then responded with its own set of changes in a cycle of influence.

Another problem, again stemming from their unilateral nature, is that the effects on incoming traffic are unpredictable. For example, the amount of traffic that will be moved by prepending is not known in advance. This forces ISPs to guess each other's routing decisions and use a "trial-and-error" [2, 10] (or, "tweak and pray") approach to manage traffic. The result is not only routing instabilities, but also that changes in network policies or topology may require restarting the trial-and-error process.

Given these difficulties, ISP operations today are costly and error-prone [18]. They rely on manual intervention and "operator-based protocols" that are neither efficient nor robust.

We call for a fresh look at how to support dynamic routing policies in the Internet via an interdomain traffic engineering (TE) architecture. We argue that the foundation of this architecture must be explicit coordination between ISPs. With explicit coordination, ISPs exchange information about their traffic and routing options, and all ISPs impacted by a potential routing change negotiate the actual change. This has three important benefits over the status quo. First, it enables ISPs to find "win-win" outcomes that benefit both relative to unilateral routing. This trivially prevents one ISP from inadvertently hurting another ISP while engineering its own network, a "win-lose" situation that can occur when ISPs are unaware of each other's resource constraints. Second, it enables predictable control over incoming and outgoing traffic, allowing ISPs to efficiently reach desired routing states. This is because an ISP is not required to guess at the effects of other ISPs' policies. Third, it makes conflicting resource policies visible so that a stable compromise can be negotiated by the involved ISPs. Today, implicit conflicts can lead to unpredictable and even unstable outcomes.

The rest of this paper is organized as follows. In §2, we motivate interdomain TE using a few examples. We discuss design considerations for an interdomain TE architecture in §3, outline the basic building blocks of a new architecture in §4, and use a simple experiment to evaluate it in §5. We conclude in §6.

† CSE Deptartment, University of Washington. Authors' e-mails: {ratul,djw,tom}@cs.washington.edu. This work was supported in part by grants from the NSF (Award ID 0133495) and Cisco Systems, and a Microsoft Fellowship.

2. MOTIVATION

We motivate interdomain TE by describing a few common activities that are difficult to accomplish today. These activities are necessitated by a basic underlying factors, such as congestion, failures, link addition or upgrade, cost concerns, and load balancing.

1. Managing inbound traffic One of the most challenging tasks today is shifting traffic between incoming links [24]. This is because of the unpredictable nature of available controls: a downstream ISP cannot predict how much traffic will move without knowing the policies of the upstream. The task is even harder for transit ISPs than for edge ISPs because these controls can affect the volume of incoming traffic itself.

2. Selecting peering points Today, peering point selection between ISPs that peer in multiple places is done unilaterally by the upstream ISP. This leads to both sub-optimal paths due to early-exit routing [29] and instabilities such as the incident mentioned in §1. No current mechanism enables ISPs to judiciously select peering points to avoid such problems. MEDs were designed to enable downstream ISPs to select peering points, but they simply replace early-exit with late-exit [29].

3. Managing outbound traffic (for transit ISPs) Managing outbound traffic is usually considered easy because BGP allows arbitrary route selections. But it is not easy for transit ISPs to do this in a stable manner. This is because changes in outbound AS paths need to be propagated upstream. Depending on upstream policies, this can unpredictably increase or decrease their incoming traffic [9].

4. Managing scheduled events Certain scheduled events significantly impact traffic patterns. Coordination for such events is highly desirable; today it has to be done manually. For instance, major organizations consult their providers as to when it is safe to do cross-campus backups. This is because, if done at an inopportune time, the traffic generated by such flows can overwhelm the provider's network. Similarly, large ISPs inform each other of their maintenance windows.

The common theme above is unpredictability, instability, and manual intervention arising from lack of coordination. Unsurprisingly, automated coordination between ISPs is the basis of our approach. We discuss how it simplifies the tasks above in §4.4.

3. DESIGN CONSIDERATIONS

In this section, we discuss the design considerations of an interdomain TE architecture. We divide them into functional requirements and operational constraints.

3.1 Functional Requirements

The fundamental goal of TE is to efficiently use network resources from a cost and performance perspective. Intradomain TE attempts to achieve this while keeping incoming and outgoing traffic pattern (traffic IO) fixed [12]. Consequently, it is not effective in many situations [9]. Interdomain TE goes a step further in that it tries to control the traffic IO itself. Thus, our primary requirement:

1. Predictable control over traffic IO An ISP should be able to control its traffic IO. This includes activities such as picking the outgoing link, increasing or decreasing the amount of incoming traffic, and moving incoming traffic from one external link to another, to optimize cost or performance or to isolate customers from each other. There are two distinct but related aspects to this control. First, an ISP should be able to modify its existing traffic IO. Predictability in this case implies that the system should move from the current to the final state without undesirable, intermediate states. The interdependence between traffic IOs of various ISPs calls for the second aspect: the changes to the traffic IO of an ISP caused by other ISPs should be predictable. This enables an ISP to plan ahead by either accommodating the change or precluding it so that it does not violate local resource policies. Contrast this with the current situation in which modifications are governed by trial-and-error [2] and unpredictable changes caused by other ISPs leads to "fire-fighting" where operators fix problems only after resource policy violations.

Predictable control, even though provided at a local (ISP) level, provides a strong basis for global stability: $i)$ the system does not shuffle through undesirable states when the external factors (e.g., topology and traffic) are stable; $ii)$ ISPs do not inadvertently violate each other's resource policies, which means that they will not get stuck in cycles of counter-changes; $iii)$ ISPs can protect themselves from the capriciousness of other ISPs that frequently change their routing.

2. Stable resolution of dynamic policy conflicts There is an inherent contradiction in the requirement above. Because the traffic IOs of various ISPs are interdependent, it is impossible to simultaneously provide absolute control to all ISPs. Thus, our second requirement is that an interdomain TE architecture recognize and stably resolve conflicting policies. Hidden, unresolved conflicts lead to unstable situations today. There are two ways to resolve conflicts. The first is to architecturally mandate a solution, such as the upstream ISP always wins, or use a deterministic conflict resolution mechanism based on inputs from the ISPs. But this approach is too rigid; it fails to account for special situations and different relationships between ISPs. For instance, in some cases provider ISPs might be willing to concede to their customers in return for additional compensation. Similarly, since these conflicts will arise multiples times, ISPs might be willing to take turns to concede. The second, and our preferred, way is to let the outcome be determined by the ISPs based on their unique situation. Thus, the architecture should be an enabler, not the arbiter, for conflict resolution [6].

3.2 Operational Constraints

Interdomain TE has to work with autonomous, competing ISPs. This introduces the two key constraints described below. (It also hinders the straightforward extension of intradomain TE methods [12] as a solution).

1. Limited information disclosure Competitive concerns make ISPs reluctant to share information about the internal state of their network with other ISPs. The interdomain TE architecture should respect this and not require that potentially sensitive information be disclosed, e.g., topology and performance data. Instead, it should be sufficient to coordinate using knowledge of the routing options and their relative desirability.

2. Accommodating varied interests While ISPs cooperate to provide overall connectivity, their interests are not completely aligned: different ISPs prefer different routing patterns for various reasons. Further, ISPs are profit-maximizing and interested in their own efficiency, rather than global efficiency. Thus an interdomain TE architecture should enable ISPs to compute mutually satisfactory solutions where each ISP is maximizing its own objective function. In game-theory parlance, negotiations should arrive at Pareto-efficient routing outcomes in which there is no other outcome that is better for all ISPs. Because ISP routing is not a "zero-sum" game, both ISPs can win in a negotiation compared to routing today (as we will see in §5). Note that Pareto-efficient outcomes improve stability because no ISP can improve its situation at a cost to others; with the absence of Pareto-efficient outcomes today, ISPs have an incentive to use ad hoc mechanisms to alter the outcome.

4. ARCHITECTURAL BUILDING BLOCKS

We outline our approach in this section by describing its basic building blocks, independent of the underlying routing protocol. We use the term *flow* to refer to a collection of packets, such as the traffic between two edge ISPs, that share the same path through the network. The terms *upstream* and *downstream* are relative to the direction of the data traffic. (Recall that in BGP routing information flows from downstream to upstream.)

4.1 Two-way Information Exchange

We argue that predictable control over traffic requires routing information exchange in both directions – from upstream to downstream and vice versa. The discussion below is in the context of a downstream ISP trying to control incoming traffic – a harder task than controlling outgoing traffic in the current Internet.

It is virtually impossible for a downstream ISP to have predictable control over its incoming traffic in a routing framework such as today's in which routing information flows only from downstream to upstream. To understand this point, note that there are only two kinds of TE information that downstreams can send upstream, neither of which satisfactorily address the problem. First, *directives* force upstreams to use particular paths. Examples of directives in the current Internet are MEDs and selective announcements. But directives cause upstreams to lose control over their outgoing traffic. Thus this only transfers the problem elsewhere instead of solving it.

Second, *suggestions* tell upstreams about downstream preferences. AS-path prepending is an example of a suggestion. Whether an upstream obeys a suggestion depends on its policies and available routing options. But since the downstream does not have this information, it can only guess the impact of a particular suggestion, forcing it to use an unpredictable, trial-and-error approach. Apart from the routing instability problem with this approach, it also requires recalibration whenever there are changes in the upstream or local network topology or policies. A time-consuming recalibration process will be particularly painful right after a failure.

The problem with suggestions is not limited to BGP mechanisms, but extends to other potential mechanisms. We discuss two such mechanisms that might otherwise seem plausible approaches to interdomain TE. Consider a transit ISP trying to control how traffic enters its network. The first mechanism is rate-limiting: the ISP monitors its incoming traffic and rate-limits flows that cause congestion. But this is not predictable because the transit ISP does not know in advance how much rate-limiting is required to discourage a particular flow. Small amounts of rate-limiting could result in large changes in load as upstreams shift traffic to uncontrolled links. Moreover, rate-limiting is a poor feedback channel; since an upstream does not know the downstream ISP's load tolerance, it cannot predict whether a particular change would lead to rate-limiting without implementing the change. The second mechanism is pricing: the ISP increases the price of carrying traffic along congested paths. But the amount of traffic that will move as a result of increasing the price by a certain amount is not predictable as the transit ISP does not know the prices of other options available to the traffic sources. Additionally, to be effective, prices must vary with load inside the transit ISP. This calls for fine-grained pricing, which to date has proven impractical [21].

Control over traffic can be made predictable through a two-way information exchange. One possibility is for upstreams to disclose their policies, resource constraints and available routing options – the bases for routing decisions. But competitive concerns rule this out. A more viable mechanism is to explicitly coordinate with the upstreams, asking them whether they are willing to make the desired routing change (possibly in return for some favor). This also makes changes more predictable for upstreams and enables them to preclude changes that violate local policies. In a similar manner, an ISP can control its outgoing traffic by coordinating with downstream ISPs to see if they are willing to accept routing changes to their incoming traffic.

4.2 Route Negotiation

In some cases, one or more ISPs will not be interested in making the change desired by an ISP because it conflicts with their own resource policies. The value of explicit coordination is that it makes these policy conflicts explicit before the policies are violated due to traffic movements. At this point, these conflicts can be stably resolved.

We propose that ISPs resolve conflicts by negotiating among themselves. The outcome of this negotiation is a set of flow routing paths acceptable to all ISPs. We make two observations regarding the negotiation process, partly inspired by economic and political negotiations [26, 4]. First, the chances of a negotiation leading to outcomes that satisfy all ISPs are better when more flows are simultaneously negotiated. This enables ISPs to concede a little on one flow for more gain on another, such that the overall gain is positive. These additional flows can be either other flows being exchanged by the negotiating ISPs or flows traversing the bottleneck resource because their movement is likely to help with the negotiation.

Second, and an extension of the above, the chances of a successful negotiation are higher if ISPs are willing to compromise in the present for future benefits.[1] This does not require a global virtual currency, but can be implemented using local accounting if ISPs often negotiate with the same set of other ISPs. We expect this to be true in the Internet where ISPs tend to exchange most of their traffic with a small set of other ISPs [31].

Devising appropriate mechanics for ISP negotiation is a subject of ongoing work. So far, we have looked at the limited case of negotiation between two neighboring ISPs [19]. The goal of this negotiation is to assign each flow to one of the multiple peering links between the ISPs. Each ISP assigns a numeric utility to each peering point and flow pair and shares this information with the other ISP. The utility captures a measure of how much the ISP prefers to route that flow using that peering link. Then, based on utilities of both ISPs, the two ISPs take turns to propose peering links for flows. This methodology leads to nearly optimal peering point selection from both a latency and overload perspective because ISPs negotiate over a set of flows; each ISP experiences minor losses for some flows and significant gains for others such that both of them improve their situation compared to unilateral routing. Extending this to multiple ISPs is an important part of our future work.

In scenarios where ISPs fail to reach an agreement, the flow uses its *default* path. In the current architecture this is mostly left up to the upstream ISP, but it is certainly not the only option. For instance, between neighboring ISPs, it could be contractually specified. Even when ISPs fail to agree, the explicit nature of conflicts ensures stability as ISPs understand the situation and will not unilaterally control the flow in contradictory ways.

4.3 Flow Registration

Flow registration is the enabler for inter-ISP coordination. Upstream ISPs register their flows with all the downstream ISPs the flow passes through. They include the following information: $i)$ the signature of the flow; $ii)$ estimated amount; $iii)$ alternative routing options; and $iv)$ expected lifetime of the flow. The signature enables the downstream to recognize this flow through its network. Specifying the lifetime is optional, and is otherwise assumed to be infinity. Registration is soft-state and is forgotten by the downstream unless refreshed.

New registrations occur when a flow changes or desires to change its path. The path of a flow can change because of either *forced* changes, e.g., due to failures, or *optimizing* changes in which the current path is intact but one of the ISPs wants to use a different path. For optimizing changes, the flows must be registered along the new path before moving the traffic. This alerts downstreams of impending traffic changes. If the change will overload a downstream ISP, it can let the upstream know. During forced changes, the path is changed simultaneously with starting flow registration to minimize the failover time. The danger with forced changes is that it may overload a downstream ISP. To protect unrelated traffic from a potential overload, the downstream should lower the priority of such unregistered flows.

The amount of traffic carried by a flow might change unpredictably (flow birth is a special case of this event). Increase in traffic volume is problematic because of overload possibilities. This can be dealt with in two ways. First, the upstream can locally shape the traffic to fit the currently registered profile until registration is updated. Second, when shaping is not a desirable, this increase should be considered a forced change: the downstream treats any amount above the existing registration as low priority until the registration is updated.

Our concept of flow registration differs from the traditional model of end-to-end QoS in its granularity. It operates on ISP traffic aggregates instead of individual end host or router flows. (ISPs can thus internally re-route flows as they deem fit without affecting registrations.) This granularity, combined with Internet characteristics, helps to makes our approach scalable in several ways. First, edge ISPs receive traffic only from a small fraction of other edge ISPs [23]. This significantly reduces total number of flows in the Internet (and no single ISP sees all the flows). Second, a small fraction of flows consume a large fraction of the bandwidth. Two studies found that roughly 10% of the ISP-level flows represent 90% of the bandwidth [31, 8]. We can leverage this skew by limiting registration to big flows. Third, it helps that the traffic carried by big flows is relatively stable over time [31, 8], implying that their registration churn would be low. Shaikh *et al.* have shown that intradomain TE is effective and stable when ISPs focus only on big, long-lived flows [28].

4.4 Examples

In this section, we briefly discuss how our approach simplifies the TE tasks of §2.

[1] This is similar to why economies are more efficient with currency than with barter: trading continues even when the needs of players are not aligned in time.

1. Managing inbound traffic Consider an ISP wishing to move a certain amount of traffic between incoming links. In our scheme, the ISP would negotiate with its upstreams to move some of the registered traffic to an alternate path. When the upstream is not directly connected to the ISP, the movement also requires permission from the intermediate ISPs. This method is helpful in predictably managing incoming traffic both during routine operations and immediately after a failure.

2. Selecting peering points Neighboring ISPs can negotiate peering points for all flows they exchange (as described §4.2). Such a negotiation also enables automated management of events such as peering link addition.

3. Managing outbound traffic (for transit ISPs) Recall that the problem with managing outbound traffic is that the incoming traffic itself might change unpredictably. But with our approach, a transit ISP can stably manage its outbound traffic because any additional incoming traffic will be registered.

4. Managing scheduled events Scheduled events can be automatically managed by registering future flows. This enables ISPs to agree on a mutually convenient time when possible and informs all impacted ISPs of impending traffic changes to help them plan better.

5. A COORDINATION EXPERIMENT

We now compare a scenario in which edge ISPs unilaterally choose their routing, regardless of its impact on transit ISPs, with one in which the edge ISPs coordinate their routes with transit ISPs. Our goal is to highlight the value of explicit coordination. Admittedly, it is but one of many possible experiments and has been considerably simplified due to space constraints. Nonetheless, the routes selected by edge ISPs are legitimate choices in the routing architecture today, and we believe the experiment suggests that coordination has potential benefits.

As input, we use measured PoP-level topologies of 60 ISPs (17 tier-1 and 43 lower tier) and their interconnections [29]. There are a total of 443 lower tier PoPs and 1145 connections between them and tier-1 ISPs. Traffic flows between all pairs of lower tier PoPs, with tier-1 ISPs providing transit as needed. We compute the amount of traffic between a pair of PoPs using the gravity model [20] which states that this amount is proportional to the product of the weight of the PoPs. We use the population of the PoP's city [1] as its weight, which yields an Internet-like skewed traffic distribution in which bigger cities source and sink more traffic [3].

We assume that transit ISPs aim to provide consistent, high quality service to their customers. This is commonly accomplished by overprovisioning [5, 30]. The resulting low link utilizations logically isolate the ISP's customers from each other and its peers. Thus, we use the required overprovisioning as our metric of cost for transit ISPs.

Without coordination, the source PoP of each flow randomly picks one of the directly connected tier-1 ISPs as the next hop. This is intended to be a simplistic model of smart routing [27, 15] – while the choices there are not random, neither are they well-matched to the policies of transit ISPs. Once traffic enters the tier-1 ISPs, it is routed to the destination using the common interdomain routing policies of early exit and shortest AS-path while respecting commercial ISP relationships [13, 29].

We measure overprovisioning with experiments that consist of a number of iterations. In each iteration, traffic is routed as described above, resulting in a randomized set of choices across iterations. We deem the necessary overprovisioning for a link to be the difference between the maximum traffic it carries across all iterations and the traffic in the first one; this is the level needed to ensure that no choices by edge ISPs lead to congestion. Using the first iteration as the reference underestimates overprovisioning compared to using the most efficient one. Different experiments have different first iterations, producing a range of overprovisioning values. The overprovisioning factor for an ISP is the sum of the provisioning level of its links normalized by the sum of traffic on its links in the first iteration. This normalization accounts for the base capacity of each link, which reflects the economic reality that doubling capacity is costlier for fatter links.

With coordination, flows require approval from all tier-1 ISPs along the path. If a particular choice is not approved, the source ISP tries a different one. Tier-1 ISPs approve a flow if it will not lead to more than a 25% increase[2] in link load compared to the first iteration (against which overprovisioning is measured). Approvals are granted on a first come first serve basis and for simplicity we do not consider retractions. To mimic negotiation failures, each flow has a default path – the one used in the first iteration – which can be used even without an approval. This model of ISP coordination is very simple; in practice, ISPs can disclose to their neighbors their internal utility for various routing paths. These utilities will be used to make mutual compromises over individual flows such that all ISPs gain in aggregate and the result is Pareto-efficient.

Figure 1 shows the results of this experiment for fifteen runs with different random seeds. Each run consisted of 200 iterations, which was sufficient to get stable results. We see that without coordination overprovisioning is likely to be very expensive: half the ISPs require more than 50% overprovisioning and a quarter of them require twice that amount. With coordination, tier-1 ISPs can significantly reduce overprovisioning: it is always close to 25% (per the approval policy). It exceeds 25% for some ISPs only because default flows can always use a link. We also considered whether intradomain TE alone can reduce overprovisioning through internal rerouting while keeping the same

[2] The choice of 25% as the threshold is arbitrary. In practice, ISPs would be free to choose their own acceptable overprovisioning level, balancing the added revenue possible from not turning away traffic against the cost of overprovisioning. For simplicity, though, we keep all ISPs at the same overprovisioning level.

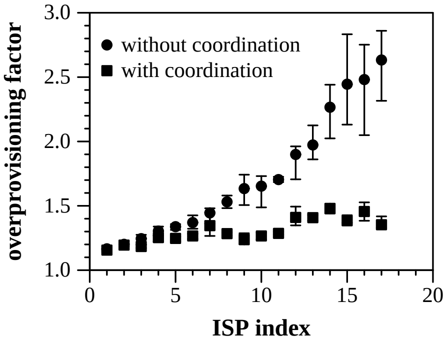

Figure 1: *Overprovisioning required at tier-1 ISPs with and without coordination. The graph plots the mean and interquartile range from fifteen experiments.*

traffic IO. We found it to be largely ineffective.

As a check that there are no substantial hidden costs with coordination, we measured how often edge ISPs received their preferred paths. 80% of the flows always got their first choice and 99.9% got one of their top two choices (with the first choice in some iterations and the second one in others). For comparison, note that 50% of the flows have two or more choices to choose from and 25% have four or more choices. Thus, flows often managed to get paths of their choice. If, as in content distribution networks [16], the benefit of smart routing stems from avoiding bad choices rather than picking optimal ones, the inability to always pick the first choice should not have a serious impact on user performance.

6. CONCLUDING REMARKS

The Internet needs a principled interdomain TE architecture to enable ISPs to stably optimize their networks. The current approach of relying on cobbled, ad hoc techniques yields neither stability nor efficiency. We outlined an architecture based on explicit coordination between ISPs to predictably control traffic and to bring policy conflicts to the fore so that they can be resolved. Implicit policy conflicts can lead to unpredictable and unstable outcomes. Results from a simple experiment with realistic ISP topologies suggest that this approach has the potential to help ISPs meet their goals without adversely impacting others.

Our work is far from complete. We have outlined a set of building blocks, not a complete architecture. Several issues remain open, including the exact nature of ISP negotiation, scalability, and secure inter-ISP communication. Another important open issue we have not explored is the interplay of negotiation with pricing and commercial contracts. We also need to consider the deployment of our architecture in the Internet, perhaps leveraging recent work on logical centralization of ISP routing [11] and network capabilities [25] for ISP communication and flow registration.

7. REFERENCES

[1] Center for International Earth and Science Information Network. http://www.ciesin.columbia.edu.

[2] D. Awduche, *et al.* Overview and principles of Internet traffic engineering. RFC 3272, IETF, 2002.

[3] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. PoP-level and access-link-level traffic dynamics in a Tier-1 PoP. In *ACM SIGCOMM IMW*, 2001.

[4] S. J. Brams. *Negotiation Games: Applying game theory to bargaining and arbitration*. Routeledge, 1990.

[5] C.-N. Chuah. A Tier-1 ISP perspective: Design principles & observations of routing behavior. http://sahara.cs.berkeley.edu/jun2002-retreat/chuah_talk.pdf, 2002.

[6] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in cyberspace: Defining tomorrow's Internet. In *ACM SIGCOMM*, 2002.

[7] D. Estrin, Y. Rekhter, and S. Hortz. Scalable inter-domain routing architecture. In *ACM SIGCOMM*, 1992.

[8] W. Fang and L. Peterson. Inter-AS traffic patterns and their implications. In *Global Internet Symposium*, 1999.

[9] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. *ACM CCR*, 33(5), 2003.

[10] N. Feamster, J. Winick, and J. Rexford. A model of BGP routing for network engineering. In *ACM SIGMETRICS*, 2004.

[11] N. Feamster, *et al.* The case for separating routing from routers. In *ACM SIGCOMM FDNA Workshop*, 2004.

[12] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communication Magazine*, 2002.

[13] L. Gao. On inferring autonomous system relationships in the Internet. In *IEEE Global Internet Symposium*, 2000.

[14] V. Gill. Private Communication, 2003.

[15] Internap. http://www.internap.com/.

[16] K. Johnson, J. Carr, M. Day, and F. Kaashoek. The measured performance of content distribution networks. In *Int'l Web Caching and Content Delivery Workshop*, 2000.

[17] K. Lougheed and Y. Rekhter. A border gateway protocol (BGP). RFC 1105, IETF, 1989.

[18] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *ACM SIGCOMM*, 2002.

[19] R. Mahajan, D. Wetherall, and T. Anderson. Interdomain routing with negotiation. Tech. Rep. CSE-04-06-02, University of Washington, 2004.

[20] A. Medina, *et al.* Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM*, 2002.

[21] A. Odlyzko. Pricing and architecture of the Internet: Historical perspectives from telecommunications and transportation. http://www.dtc.umn.edu/~odlyzko/doc/pricing.architecture.pdf, 2004.

[22] BGP reports. http://bgp.potaroo.net/.

[23] B. Quoitin, S. Uhlig, and O. Bonaventure. Using redistribution communities for interdomain traffic engineering. *Quality of Future Internet Services*, 2002.

[24] B. Quoitin, *et al.* Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 41(5), 2003.

[25] B. Raghavan and A. Snoeren. A system for authenticated policy-compliant routing. In *ACM SIGCOMM*, 2004.

[26] H. Raiffa. *The art and science of negotiation*. Harvard University Press, 1982.

[27] RouteScience. http://www.routescience.com/.

[28] A. Shaikh, J. Rexford, and K. Shin. Load-sensitive routing of long-lived IP flows. In *ACM SIGCOMM*, 1999.

[29] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, 2003.

[30] T. Telkamp. Traffic characteristics and network plannning. NANOG, http://www.nanog.org/mtg-0210/ppt/telkamp.pdf, 2002.

[31] S. Uhlig and O. Bonaventure. Implications of interdomain traffic characteristics on traffic engineering. *European Transactions on Telecommunications*, 2002.