Routing as a Service

Karthik Kalambur Lakshminarayanan
Ion Stoica
Scott Shenker
Jennifer Rexford

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-19
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-19.html

February 27, 2006

Routing as a Service

Karthik Lakshminarayanan Ion Stoica Scott Shenker Jennifer Rexford
University of California, Berkeley Princeton University

Abstract

In Internet routing, there is a fundamental tussle between the end users who want control over the end-to-end paths and the Autonomous Systems (ASes) who want control over the flow of traffic through their infrastructure. To resolve this tussle and offer flexible routing control across multiple routing domains, we argue that customized route computation should be offered as a *service* by third-party providers. Outsourcing specialized route computation allows different path-selection mechanisms to coexist, and evolve over time.

1 Introduction

Interdomain routing has long been based on three pillars:

- *Local control*: ASes have complete control over routing and forwarding decisions within their domain.
- *Bilateral agreements*: An AS has pairwise contracts with neighboring ASes to collaborate in providing service.
- *Distributed algorithms*: End-to-end paths are computed using a distributed routing protocol, where each AS applies local policies to routes learned from neighbors.

The distributed routing protocols produce default paths that satisfy most customers. However, the default paths are not sufficient for some customers with special performance or policy requirements. In this paper, we propose that third-party Routing Service Providers (RSPs) satisfy the needs of these customers through (i) end-to-end control over the forwarding infrastructure, (ii) business agreements with the various ISPs along the paths, and (iii) logically-centralized computation of the paths based on a global view of the topology.

1.1 Tussle Between Users and ISPs

In today's routing architecture, end-to-end path selection depends on the complex interaction between thousands of ASes, ranging from Internet Service Providers (ISPs) to enterprise networks. Each AS has control over the flow of traffic through its part of the infrastructure and cooperates with neighboring ASes to select paths to external destinations. The operators of these ASes configure the routing protocols running on their routers to make efficient use of network resources, maximize revenue in sending traffic to customers, and control which neighbors can transit traffic through their infrastructure. Still, each ISP has at best indirect control over the end-to-end path, typically by "tweaking" the routing-protocol configuration, making it difficult to offer meaningful service-level agreements (SLAs) to customers or to identify the AS responsible for end-to-end performance problems.

An ISP's customers, such as end users, enterprise networks, and smaller ISPs, have even less control over the selection of end-to-end paths. By connecting to more than one ISPs, an enterprise can select from multiple paths [2]; however, the customer controls only the first hop for outbound traffic and has (at best) crude influence on incoming traffic. Yet, some customers need more control over the end-to-end path, or at least its properties, to satisfy performance and policy goals. For example, a customer might not want his Web traffic forwarded through an AS that filters packets based on their contents. Alternatively, a customer might need to discard traffic from certain sources to block denial-of-service attacks or protect access to a server storing sensitive data. Another customer might want low end-to-end delay for Voice-over-IP traffic, or high throughput for a large data transfer.

The conflict between ISPs and their customers for control over path selection is a fundamental "tussle" [5]. Unfortunately, existing proposals skew the control to one stakeholder at the expense of the other. On the one hand, ubiquitous deployment of a QoS-routing protocol would enable ISPs to select end-to-end paths that satisfy user requirements. However, QoS routing between ASes would require deploying a complex protocol that is needed for only a small fraction of requests, and even then is unlikely to meet all the specialized needs. On the other hand, source routing would give customers complete end-to-end control over the forwarding paths. However, ISPs do not have an economic incentive to cede control over routing decisions, due to the lack of business relationships with end users; in addition, source routing introduces difficult scalability and security challenges. Instead, we argue for pulling the tussle out of the infrastructure by allowing third-party providers to form business relationships with both users and ISPs, and select and install end-to-end forwarding paths on behalf of the users.

1.2 Routing as a Service (RAS)

Our proposal consists of three entities: the forwarding infrastructure (FI) that spans multiple underlying ASes, a collection of Routing Service Providers (RSPs), and the clients of the RSPs, as illustrated in Figure 1. The RSPs contract with both ASes and end-customers, so they do not have to negotiate directly. More specifically, we envision an RSP would buy *virtual links* (VLs) from various ASes with well-defined SLAs

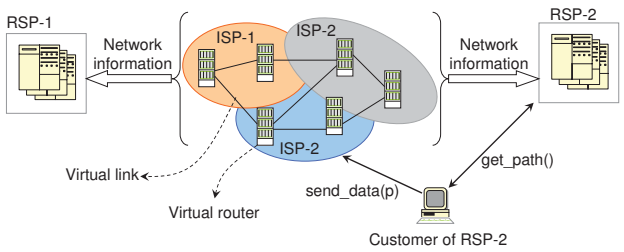

Figure 1: The main components of the RAS architecture: the forwarding infrastructure (FI), one or more routing service providers (RSPs), and RAS clients.

(something ISPs are quite willing to sell today), connecting some number of *virtual routers* (VRs). The RSP sets the forwarding state of these VRs, though the underlying ASes control how traffic flows between VRs. Customers desiring customized routes contract with an RSP, which would then set up an appropriate end-to-end path along its virtual links. The fact that there are a limited number of these VRs allows the RSP to compute these routes in a centralized fashion, so that the path characteristics can be carefully tuned. Multiple RSPs may coexist, forming a competitive market-place for offering a value-added service to customers at a reasonable price.

Our architecture meets the needs of both parties. The ASes still have sufficient control, in that they can limit the number and size of the VLs they sell and engineer the flow of traffic through their networks. The end-customers get the path performance they need, and do not have to deal with every AS along the path. Our approach has precedence, in that Content Distribution Networks (CDNs) perform a similar role. Rather than having content providers contract with every ISP for caching, and having some complicated inter-ISP protocol for deciding who serves which requests, a CDN acts as a middleman in the process. The CDN has contracts with the ISPs, and then is able to offer a comprehensive content delivery service for content providers. Most Web sites do not need a CDN so, rather than complicate the basic HTTP protocol with sophisticated content-delivery mechanisms, only those customers with specialized needs contract with a CDN. As with CDNs, we envision that a few competing RSPs would coexist and provide flexible service for different customers.

In the next section, we present three examples of customer requirements that RSPs could satisfy, overcoming fundamental limitations with today's routing architecture. Section 3 discusses the interaction between ISPs and RSPs, and Section 4 explores how customers interface to the RSP. Section 5 concludes with a discussion of future research directions.

Related work: Though several proposals give end-hosts more control over routing, most do not directly address the tussle between ISPs and customers. We list only the most related prior work here. To promote competition among providers, Yang [13] proposes a solution that allows both senders and receivers to choose routes at the AS level. The Nimrod [4] architecture proposed computation of routes by the clients of the network, and introduced mechanisms for distribution of network maps. Broker [3] is a centralized entity that computes routes based on QoS requirements within a domain, and across domains. None of these approaches provide flexible, yet scalable, mechanisms for selecting end-to-end paths that RAS aims to provide. The RCP proposal [6] advocates moving control of routing from the individual routers to dedicated servers in each AS, but does not consider giving third-party providers control over the end-to-end path.

2 Case for End-to-End Route Control

Although default routes are sufficient for most traffic, some traffic needs to follow paths that satisfy high-level policy goals. In this section, we present three examples that illustrate why flexible route control is important, how today's routing architecture is insufficient, and how Routing Service Providers (RSPs) can direct traffic on the appropriate paths.

2.1 Example 1: Avoiding Undesirable ASes

Avoiding paths through certain ASes: Some users may want their traffic to avoid traversing certain intermediate ASes. For example, suppose an AS is known to perform content-based filtering of data packets, or to redirect Web traffic to alternate Web servers that return sanitized content. Alternatively, consider two government agencies that would not want their traffic to be traverse networks run by another country. Or, suppose that a user wants its traffic to avoid ASes that do not apply best common practices for securing the router infrastructure or preventing DDoS attacks. In each of these cases, the end user wants the packets to and from certain destinations to avoid forwarding paths that traverse particular ASes.

Avoiding selected ASes is easier with RAS: Today, path selection depends on the composition of the BGP routing policies implemented in multiple ASes. Selecting paths that avoid certain ASes is difficult, if not impossible; RAS can overcome these limitations:

- Because BGP is a path-vector protocol, an AS only learns the paths advertised by its immediate neighbors. If all of these paths traverse an undesirable domain, the AS has no way to select a suitable path, even if such paths exist in the AS graph. In RAS, an RSP that has complete information about the virtual topology can easily compute paths that avoid selected ASes.
- BGP routers select a single best path for each destination prefix. This precludes an ISP from allowing one customer to avoid a downstream AS (for policy reasons) while allowing other ASes to use paths that traverse the AS (e.g., for performance reasons). In RAS, an RSP can selectively direct some traffic to a special path that avoids the selected AS, while allowing the remaining traffic to use the default path.

- Today, ASes use BGP routing policy to implement business relationships with neighboring domains. The ISP does not have an economic incentive to direct traffic through a peer or provider, even if the path avoids the AS in question, if a path exists through one of its customers. In RAS, the RSP has its own business relationship with the ISP, which provides the necessary incentive for the ISP to direct selected traffic on the chosen path.
- BGP is destination-based, making it extremely difficult to ensure that the reverse path from the destination back to the source avoids the AS in question. In RAS, the RSP can install forwarding state along both directions of the path between the two hosts.

That said, the RSP (or set of RSPs) must resolve potential conflicts between the desires of the sending and receiving hosts. For example, the sender might want to avoid a particular AS, whereas the receiver might prefer paths that traverse this AS. We argue that the RSPs are the natural place to resolve these inherent tensions, based on full knowledge of the virtual topology and the routing policies of each party.

2.2 Example 2: Blocking Unwanted Traffic

Discarding traffic from unwanted senders: End hosts may want control over which sources can send traffic to them, and which links they can use. For example, the victim of a denial-of-service attack may want to block the offending traffic, based on the source IP address and where the traffic enters the network. To prevent future DoS attacks, an enterprise might block traffic from source prefixes in geographic regions known for being the source of attacks. Similarly, to prevent spam, an enterprise may want to block traffic from the IP addresses of mail servers known for sending spam. A university campus may want to block incoming traffic with a source port number corresponding to certain application. Sites in a virtual private network (VPN) might want to receive traffic only from addresses belonging to other sites in the VPN. In each of these cases, the end host wants the network to selectively discard incoming traffic.

Blocking unwanted traffic is easier with RAS: Today, blocking unwanted traffic depends on configuring access control lists (ACLs) or null routes at various points inside the network. Ideally, unwanted packets should be discarded close to the sender, to reduce the bandwidth consumed by the traffic and to amortize the overhead of applying the filtering rules [1]. Achieving this goal in today's routing system is difficult, but RAS can overcome these challenges:

- Today, blocking unwanted traffic depends on the joint configuration of the routing protocols and access-control lists [12]. In RAS, an RSP can forward traffic to "null" based on a wide variety of policies, such as a five-tuple of source and destination prefix, source and destination port numbers, and protocol.
- Pushing the filters further away from the target network requires cooperation between many pairs of ASes; with n ISPs, this may require $O(n^2)$ business relationships. In RAS, an RSP forms the relationships with the n ISPs to install the needed forwarding state, obviating the need for pairwise relationships.
- Knowing which traffic to block requires keeping track of the IP addresses that often originate spam (and other unwanted traffic), and knowing the local filtering policies of each destination. Having each AS maintain this information is inefficient. In RAS, an RSP can keep track of the filtering rules and install them at the relevant locations in the forwarding infrastructure.

That said, the RSP must resolve conflicts between sources who want to send packets and destinations who do not want to receive them, and balance the trade-off between dropping the traffic close to the senders and installing a large amount of state in the forwarding infrastructure. Again, we argue that RSPs are a natural place to address these trade-offs.

2.3 Example 3: Guaranteeing Quality of Service

Providing performance guarantees for traffic: Communicating hosts may want guarantees on the quality-of-service for certain traffic. For example, a user may want strict delay guarantees for interactive phone calls; a remote user listening to an audiocast may have much looser performance requirements. Another user may want a bandwidth guarantee for downloads from a video-on-demand server. Two scientific organizations may want a bandwidth guarantee for bulk transfer of a large data-set. In each of these cases, the end hosts have a particular "flow" that requires an end-to-end guarantee on one or more performance metrics.

Guaranteeing QoS is easier with RAS: Today, ISPs provide coarse-grained service-level agreements, only for traffic that stays inside a single AS. Providing fine-grained quality-of-service over an end-to-end path is extremely difficult, but RAS can address this challenge:

- Today's Internet does not provide end-to-end signaling to reserving resources along a path that traverse multiple institutions, making it difficult to offer performance guarantees. In RAS, RSPs negotiate strict QoS guarantees with individual ISPs, and then stitch virtual links together to provide end-to-end QoS to customers.
- Although an ISP can provide guaranteed QoS for highly-aggregated traffic, offering performance guarantees for individual flows is extremely challenging, in terms of the signaling overhead and the need for fine-grained packet scheduling. In RAS, RSPs reserve bandwidth across an ISP for aggregated traffic and manage the division of these resources across individual flows.
- Today's ISPs can determine which traffic should receive priority service based on bits in the packet headers.

However, an ISP cannot easily classify packets based on finer-grained information, or direct packets on different paths based on their performance requirements. In RAS, RSPs can classify packets based on diverse customer policies and assign a sequence of virtual links with the necessary performance properties for each flow.

In the absence of a standard signaling protocol for specifying requirements and reserving end-to-end resources, each RSP can decide what performance guarantees to offer, and how. This provides an opportunity for an RSP to differentiate itself by offering special QoS services to customers.

3 Virtual Links: ISP-RSP Interaction

Rather than controlling the entire forwarding infrastructure, an RSP contracts with ISPs for *virtual links* with Service-Level Agreements (SLAs). Virtual links allow ISPs to retain control over the flow of traffic within their networks, while reducing the overhead for RSPs to compute end-to-end paths.

3.1 Virtual Links With Service-Level Agreements

RSPs do not need control over packet forwarding at the level of individual routers and links, and ISPs may not be willing to cede such fine-grained control anyway. Instead, we envision that ISPs offer *virtual links* as a service that RSPs can purchase; then, the RSP constructs an end-to-end path by stitching together a collection of *virtual links* from the source to the destination. The virtual link is unidirectional, and connects two virtual routers that the RSP controls. More specifically, the virtual router could be inside the ISP network as a RSP-specific context located on the ISP's own router, or a separate network element outside the ISP but connected directly to the ISP. Either way, at each virtual router, we envision complete isolation between the forwarding state and virtual links controlled by different RSPs—hence, a misconfigured RSP cannot affect other RSPs or the ISP itself.

An ISP can use existing technologies, such as MPLS [9], to create virtual links and provide the necessary bandwidth isolation. The ISP can offer an SLA for the virtual link. On one extreme, a virtual link could be a constant-bit-rate pipe with a maximum propagation delay, allowing the RSP to construct end-to-end paths with hard QoS guarantees. On the other extreme, a virtual link could offer best-effort service, allowing the RSP to construct low-cost paths that obey the end-user's policy requirements, such as avoiding certain intermediate ASes. In some sense, virtual links are not much different than the services ISPs can offer today to direct customers; the main power lies in the ability of an RSP to stitch together virtual links across different providers.

When buying a virtual link with a particular SLA, the RSP guarantees not to exceed the maximum traffic load, though the ISP could install traffic shapers in the data plane to enforce these limits. The predictability of the offered load should greatly simplify how the ISP does traffic engineering. Whereas ISPs today must measure or infer the "traffic matrix," the virtual links can provide an upper bound on the traffic between two virtual routers. This aids the ISP in configuring the intradomain routing protocols to make efficient use of network resources and to ensure that the SLA is met even if internal failures occur. For example, the ISP would have an incentive to over-provision the network or provide explicit back-up paths, to avoid incurring a penalty when the SLA is violated. By controlling both ends of the virtual link, the RSP is in a good position to identify when these SLAs are violated, and to identify which "hop" in the end-to-end path is responsible for a performance problem.

To further improve the robustness of virtual links, neighboring ASes could cooperate to offer a virtual link that spans their networks. For example, two ISPs that peer in multiple locations could coordinate to balance load across these links, perhaps using the negotiation scheme discussed in [7]. Working together, these ISPs could mask the effects of a failure of one of the links or routers between them, allowing them to offer stronger SLAs for these "long haul" virtual links. The RSPs benefit directly from the reduced overhead of managing the virtual routers, and the higher service guarantees they can offer to end users. In fact, the presence of RSPs provide meaningful incentives for neighboring ISPs to cooperate in this manner, by paying a higher price for virtual links that span two or more ISPs.

Still, an RSP must be able to react when virtual links fail. In some cases, virtual links may fail due to planned maintenance in the ISP network. Because of their direct business relationship, the ISP can notify the RSP in advance of planned maintenance, to allow the RSP to migrate the end-user traffic to an alternate path, perhaps using a virtual link through another ISP, without violating the SLA offered to the end-user. This kind of graceful end-to-end rerouting is extremely difficult today, due to the lack of business relationships between end users and intermediate ASes. In other cases, an unexpected failure occurs. The virtual router connected to the failed virtual link must direct traffic to an alternate virtual link, or notify the RSP to compute a new end-to-end path.

3.2 Scalable Computation of End-to-End Paths

The abstraction of a virtual link plays an important role in reducing the path-selection overhead and how often RSPs must recompute the paths (e.g., due to virtual-link failures). An RSP computes paths on a virtual topology consisting of virtual routers and the virtual links between them. For an initial estimate, suppose an RSP has a virtual router for every border router in every AS, and a virtual link for each pair of virtual routers connected to the same AS and for each connection between neighboring ASes. Although the Internet consists of around $20,000$ ASes, around 80% are stub ASes [10] that have just one or two border routers. The Internet has around twenty tier-1 ISPs [10] that have around 500 border routers.

Focusing on these ISPs alone, the RSP would need to manage around ten thousand virtual routers (20×500) and five million virtual links $20 \times 500 \times 500$). The actual number, when including the other ASes, might easily grow to a few hundred thousand virtual routers and several million virtual links.

As with any large network, we adopt the conventional technique to achieve scalability—hierarchical organization. Several natural scaling techniques can reduce the number of virtual routers and virtual links substantially. For example, most ISP backbones consist of a relatively small number of Points-of-Presence (PoPs) in key cities. A large ISP with 500 border routers might have just 30 PoPs. Having a single virtual router per PoP would reduce the number of virtual routers and virtual links substantially. Considering 20 large ISPs with 30 PoPs each, the number of virtual routers drops to 600 (20×30), and the number of virtual links drops to $18,000$ ($20 \times 30 \times 30$), a much more manageable number. If some ASes cooperate to provide virtual links that span multiple networks, these number would drop even lower. With the high-end PCs available today, computing (say) shortest paths on a graph of this size is in the realm of possibility, especially with the use of incremental algorithms for path computation.

In addition, an RSP can divide the responsibility for path selection into multiple servers, each representing a particular region of the Internet. We envision hierarchical routing in an RSP to be simpler than today's Internet routing hierarchy because our division of computation is driven *only* by scalability concerns, not differing routing policies or regions of administrative control. Geographic boundaries offer a natural way to distribute the computation, for two main reasons. First, these boundaries have relatively small bisections, reducing the impact of subdividing the computation. For example, a relatively small number of (high-bandwidth) links interconnect the U.S. and Europe. The failure of a link inside the U.S. is not likely to affect how traffic flows to Europe. Second, having separate RSP servers in each major geographic regions is important for rapid responses to virtual-link failures. In fact, we envision that the local RSP servers can often react to a failure by performing a "local reroute" over just a portion of the path, rather than requiring a complete change in the end-to-end path.

4 Gateways: RSP-Customer Interaction

Customers need a way to subscribe to a particular RSP and specify the policy requirements that should drive end-to-end path selection. In addition, the data packets need a way to use the customized paths, without giving arbitrary end users direct control over the forwarding infrastructure. To address these issues, we introduce the concept of *RSP gateways*. An RSP gateway is software controlled by the RSPs which *bridges* the customers to the forwarding infrastructure. Both control -lane messages (policy specifications), as well as data-plane traffic (data packets) go through the gateway.

4.1 Control Plane: Route-setup based on Customer Preferences

A customer communicates its preferences to the RSP gateway, and the RSP gateway contacts the appropriate RSP nodes that perform the path computation and obtain a path that the customer's traffic can use. The RSP gateway explicitly performs the route setup on behalf of the customer to install the forwarding state in the virtual routers, or add a source route to the customer's data packets. By enforcing that virtual routers accept control messages only from RSP gateways, the RSP can ensure that malicious users cannot cause any damage (by inserting arbitrary paths or spoofing RSP packets). We now discuss various issues relating to how customers express their preferences to RSPs.

Customer configuration: When a customer signs up with an RSP, the RSP allows the customers to connect to a particular gateway (or a small set of gateways). The location of the RSP gateways could differ depending on the customers. For example, an enterprise that obtains service from an RSP can have an RSP gateway at the edge of its network so that its traffic is routed through the RSP. Configuration is easy, since individual users need not request paths separately. Also, policies might be decided by the enterprise and not the individual users. An alternate category of customers include end-users who want to use a particular RSP from, say, their laptop. The end-user could be an employee of the government and might want to avoid certain ASes in the path. In this case, the policy should stay with the laptop. Alternatively, end-user might want to leverage the fact that the laptop has multiple interfaces. In such cases, the gateway software running on the end-user's machine can monitor the performance of the last hop and use the link that does a better job satisfying the requirements.

Customer policy specification: Customers have to specify to the RSPs what their policies and preferences are for their specialized paths. Defining a flexible API that captures the different forms of customer preferences is required. Furthermore, in order to allow customers to flexibly use multiple RSPs, and for the different RSPs to coordinate the requests of their customers, a uniform interface across RSPs is also needed. Addressing this issue is an interesting avenue for future work; a possible area of exploration is using logical predicates [8].

Resolving conflicts. Different RSP customers might have conflicting policies when requesting for paths between one another. Since RSPs have global information, they are the natural point to resolve conflicts between senders and receivers. In order to combine the policies from different customers, the RSPs might need more information from customers as to what are the fallback policies when the primary policy cannot be satisfied.

A more general, and perhaps more challenging, form of conflicts arises from the fact that RAS architecture allows an open competitive market for multiple RSPs to coexist. Since

customers getting service from different RSPs need to talk to each other, the RSPs must coordinate to satisfy the customer requests. Though this problem is hard, it is important to note that this problem is fundamental. Unlike the Internet today, RAS creates a playing field for different customers to interact through their RSPs and resolve the conflicts.

4.2 Data Plane: Forwarding Customer Traffic and Bookkeeping

Based on the paths computed using customer policies, the RSP forwards the customer traffic. The gateways keep track of the amount of traffic sent as well as the number of specialized route requests made by the customers. The gateway rate-limits of customer traffic so that the RSP does not exceed the traffic contracts on the virtual links with the ISPs. Also, since the gateways know the complete customer statistics, billing the customers is also easy. We expect that for large-scale RSP deployments, the cost of RSP gateways can be amortized across different RSPs. We can leverage the virtual router architecture [11] to isolate the RSPs from one another within the gateway.

In turn, a customer can monitor the service it receives from the path that the RSP provides. If the guarantees do not conform to what the RSP promises, it can inform the RSP, based on which the RSP can investigate the problem.[1] In addition, the RSP can refund the customer for the lower service, and reclaim the cost from the appropriate virtual link provider. In today's Internet, such diagnosis is almost impossible—if a host receives bad service, it is extremely difficult to pinpoint the problem to a particular AS, even if the problem is just a few hops away. RSP gateways can themselves monitor the status of its customers and detect whether the ISPs do provide the SLAs they guarantee.

5 Conclusion

In order to resolve the fundamental tussle of routing control between ISPs and customers, we propose that customized route computation should be offered as a *service* by third-party providers. Outsourcing specialized route computation allows different path-selection mechanisms to coexist, and evolve over time.

While the overall approach of RAS is promising, there are particular issues that merit further research. For instance, in our basic design, RSPs have to sign SLAs with ISPs for individual virtual links. This might introduce scalability concerns when RSPs deal with tens of thousands of virtual links. Perhaps, combining various virtual link SLA into a topology-level SLA across all virtual links within a domain is a possible strategy for RSPs. Finally, we plan to investigate incremental deployment strategies for RSPs to deploy the forwarding infrastructure.

[1] Of course, we believe that if the service is popular, competition would lead to multiple RSPs, and hence users would switch to better providers if they receive prolonged bad service from an RSP.

References

[1] K. Argyraki and D. R. Cheriton. Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks. In *Proc. USENIX Annual Technical Conference*, 2005.

[2] J. Bartlett. Optimizing Multi-homed Connections. *Business Communications Review*, 32(1):22–27, January 2002.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475, 1998.

[4] I. Castineyra, N. Chiappa, and M. Steenstrup. The Nimrod Routing Architecture. RFC 1992, 1996.

[5] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. In *Proc. ACM SIGCOMM*, 2002.

[6] Nick Feamster, Hari Balakrishnan, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. The Case for Separating Routing from Routers. In *Proc. Future Directions in Network Architecture*, August 2004.

[7] Ratul Mahajan, David Wetherall, and Thomas Anderson. Negotiation-based routing between neighboring ISPs. In *Proc. Networked Systems Design and Implementation*, May 2005.

[8] T. Roscoe, S. Hand, R. Isaacs, R. Mortier, and P. Jardetzky. Predicate Routing: Enabling Controlled Networking. In *Proc. Workshop on Hot Topics in Networking*, 2002.

[9] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, January 2001.

[10] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, June 2002.

[11] J. Turner, T. Anderson, L. Peterson, and S. Shenker. Virtualizing the Net: A strategy for network de-ossification. `http://www.arl.wustl.edu/~jst/talks/hotI-9-04.pdf`.

[12] Geoffrey Xie, Jibin Zhang, David Maltz, Hui Zhang, Albert Greenberg, Gisli Hjalmtysson, and Jennifer Rexford. On static reachability analysis of IP networks. In *Proc. IEEE INFOCOM*, March 2005.

[13] X. Yang. NIRA: A New Internet Routing Architecture. In *Proc. Future Directions in Network Architecture*, 2003.