Network-Wide Decision Making: Toward A Wafer-Thin Control Plane[*]

Jennifer Rexford, Albert Greenberg, Gisli Hjalmtysson
{jrex,albert,gisli}@research.att.com
AT&T Labs–Research

David A. Maltz, Andy Myers, Geoffrey Xie, Jibin Zhan, Hui Zhang
{dmaltz,acm, geoffxie,jibin,hzhang}@cs.cmu.edu
Carnegie Mellon University

Abstract

We argue for the refactoring of the IP control plane to provide direct expressibility and support for network-wide goals relating to all fundamental functionality: reachability, performance, reliability and security. This refactoring is motivated by trends in operational practice and in networking technology. We put forward a design that decomposes functionality into information *dissemination* and *decision* planes. The decision plane is formed by lifting out of the routers all decision making logic currently found there and merging it with the current management plane where network-level objectives are specified. What is left on each router is a wafer-thin control plane focused on information dissemination and response to explicit instructions for configuring packet forwarding mechanisms. We discuss the consequences, advantages and challenges associated with this design.

1. Introduction

Despite the early design goal of minimizing the state in network elements, tremendous amounts of state are distributed across routers and management platforms in today's IP networks. We believe that the many, loosely-coordinated actors that create and manipulate the distributed state introduce substantial complexity that makes both backbone and enterprise networks increasingly fragile and difficult to manage. In this paper, we argue that the current division of functionality across the data, control, and management planes is antithetical to the desire for network-wide control. Instead, we advocate moving the *decision logic* for running the network from the individual routers into the management system. In our framework, the routers simply *disseminate* timely information about the network and respond to explicit instructions for configuring the packet forwarding behavior.

We argue that our approach will significantly reduce the complexity of IP routers while making the resulting network easier to manage. We first describe the status quo, and then present and contrast our design. Then, we give concrete examples of how operators are forced to run their networks today and how they could be better served by our design, and we end by considering the challenges facing our design.

1.1 Today's Data, Control, and Management Planes

State distributed across interconnected routers defines how a network "works." Yet, our understanding of how this state is created and maintained (and, perhaps more importantly, how it *should* be created and maintained) is surprisingly limited. Just as great care went in to splitting the Internet's functionality between the smart edge devices (such as end host computers) and the "dumb" core devices (such as routers), we need to revisit the separation of functionality between the three "planes" that affect the operation of an IP network:

- **Data plane:** The data plane is local to an individual router, or even a single interface card on the router, and operates at the speed of packet arrivals. For example, the data plane performs packet forwarding, including the longest-prefix match that identifies the outgoing link for each packet, as well as the access control lists (ACLs) that filter packets based on their header fields. The data plane also implements functions such as tunneling, queue management, and packet scheduling.
- **Control plane:** The control plane consists of the network-wide distributed algorithms that compute parts of the state in the data plane. For example, the control plane includes BGP update messages and the BGP decision process, as well as the Interior Gateway Protocol (such as OSPF), its link-state advertisements (LSAs), and the Dijkstra's shortest-path algorithm. A primary job of the control plane is to compute routes between IP subnets, including combining information from each routing protocol's Routing Information Base (RIB) to construct a single Forwarding Information Base (FIB) that drives packet forwarding decisions.
- **Management plane:** The management plane stores and analyzes measurement data from the network and generates the configuration state on the individual routers. For example, the management plane collects and combines SNMP (Simple Network Management Protocol) statistics, traffic flow records, OSPF LSAs, and BGP update streams. A tool that configures the OSPF link weights and BGP policies to satisfy traffic engineering goals would be part of the management plane. Similarly, a system that analyzes traffic measurements to detect denial-of-service attacks and configures ACLs to block offending traffic would be part of the management plane.

In today's IP networks, the data plane operates at the timescale of packets and the spatial scale of individual routers, the control plane operates at the of timescale of seconds with an incomplete view of the entire network, and the management plane operates at the timescale of minutes or hours and the spatial scale of the entire network.

In this paper, we argue that this three-level division of functionality leads to complex decision logic split across multiple

[*] Research sponsored by the NSF under ANI-0085920 and ANI-0331653. Views and conclusions contained in this document are those of the authors.

entities, with a mixture of manual and automatic updates required to maintain different kinds of network state. The state consists of:

- **Dynamic state:** Some state is dynamically collected and calculated through the operation of the routing protocols in the control plane (e.g., interface up/down status, OSPF link-state database, BGP process updating local BGP RIB, and the FIB itself).
- **Configuration state:** Other state is codified in the configuration commands used to program the routers' control plane (e.g., OSPF link weight and BGP import policy) and data plane (e.g., ACLs and RED parameters).
- **Hard-wired state:** Still other state is hard-wired in the router software, as default values or parameters (e.g., routing protocol timers and default RED parameters) or algorithms (e.g., Dijkstra's shortest path computation and the logic for merging multiple RIBs into a single FIB).

Although changes to a router's state (e.g., installing an ACL or changing an OSPF weight) have an impact on network-wide reachability, we have no framework that can predict how local configuration decisions affect network-wide behavior.

1.2 Cobbling Together a Network-Wide View

The past several years have seen a growing awareness in networking research and operational communities that the control and management of IP networks must fundamentally be driven by *network-level* information. That is, timely, accurate, network-wide views of topology, traffic, events and anomalies are needed to run a robust IP network. For example,

- **Traffic engineering and planned maintenance:** Maintaining stable, predictable performance during localized disruptions (e.g., failures or planned maintenance activities at the optical layer), requires a network-level understanding of the traffic matrix—the observed volume of traffic from each ingress point to each egress point. Armed with that knowledge and with detailed views of network-level topology, the operator is in a position to manipulate routing to survive the disruption, trading off performance, policy, and economic goals.
- **Centralized router configuration:** To increase stability and protect against configuration errors, changes to network elements are increasingly being funneled through a few interfaces; e.g., a small set of automated tools and/or a very small, expert team. For example, an operator might use scripts that detect inconsistency among the ACLs and blackhole routes that form the network's perimeter defense to prevent manual provisioning changes from accidentally opening up a security vulnerability.

Yet the division of functionality between the control and management planes has not evolved in any significant way in the face of these trends toward network-level management. Instead, network operators must manipulate commands at the router-level to *indirectly* enforce network-level abstractions and constraints on operational behavior. Rather than *retrofitting* network-level views and controls on top of the existing network — which frequently devolves into essentially "robotizing" the process of typing at the command-line interface of the routers — we claim the set of functions on each router must be changed to *directly* support the network-level abstractions needed for automated, network-wide management. In this paper, we argue that the current division of functionality between the data, control, and management plane has resulted in uncoordinated, decentralized state updates that are antithetical to the goals of network-wide decision-making. We advocate that *network-level expressibility* should be a first-order principle in deciding where (and whether) to place state and logic in the network.

2. Refactoring the IP Control Plane

In this section, we argue that the IP control plane should be refactored so that routers primarily forward packets and disseminate information, with a new decision plane created from a synthesis of the decision logic lifted out of the routers and the current management plane. We discuss how our proposal is the logical extension of the trends toward the use of measurement data in running the network, the evolution of clear interfaces for routing software and hardware, and the movement of path computation from routers to separate servers.

2.1 Tomorrow's Dissemination and Decision Planes

We argue that the architecture of Internet control and management should be driven by a principle of *network-level expressibility*. That is, the architectural intent and operational constraints governing the network should be expressed directly, and then automatically (via protocols or programmatic interfaces) translated to assign roles and functionality to individual routers. Until this occurs, we expect the design and operation of robust IP networks to remain a difficult challenge, and the state of the art to remain a losing battle against a trend where ever more rich and complex configuration state and logic is exposed primarily through router-level interfaces, and designed for router-by-router manual operation. While network-level expressibility can be realized in a number of ways, we focus on a design where IP control functionality is factored into a dissemination plane and a decision plane:

- **Dissemination plane**: The dissemination plane's primary objective is the timely, reliable dissemination of information to and from the network elements.
- **Decision plane**: The decision plane's primary objective is to make *all* decisions driving network behavior, including reachability, routing, access control, security, and interface configuration.

By separating information distribution and decision, we enable network-level expressibility. The dissemination plane provides the information needed to create a network-wide view. The decision plane, which contains all the network-level configuration state and logic that has been elevated outside of the routers, uses this view to directly compute the desired data plane state, including the FIB entries, packet filters, etc. The dissemination plane then distributes this data plane state to the routers that act on it.

As a result, the control plane on the individual routers becomes wafer-thin. This has several technical advantages:

- **Reducing the complexity of the routers:** Today's routers integrate an enormous amount of complex logic. Our approach reduces the autonomy of routers, making them devices that report measurement data and accept instructions that dictate the behavior of the data plane.

- **Avoiding replication of state and logic:** Today's management plane must replicate the state and logic from the control plane so that it can predict how the control plane will react to configuration changes it makes as it tries to steer the network towards meeting its goals. Our approach makes the decision plane the sole location where this information and logic reside.
- **Direct control over the network:** In today's IP networks, the management plane has (at best) indirect control over the data plane on the routers by manipulating the configuration parameters. In our approach, the decision plane has direct control over the data plane, making it easier to satisfy high-level goals.

Although our approach can improve the management of today's data plane, we believe that the data plane should evolve over time to support network-wide decision logic. For example, the data plane could support the following:

- **Unified forwarding logic:** The data plane could provide a forwarding paradigm that integrates packet filtering, address transformation, and packet forwarding, to allow the decision logic to directly specify the handling of packets (e.g., packet forwarding based on the five-tuple of the source and destination, port numbers, and protocol and efficient support for policy-based routing). This would allow the decision plane to have direct control over all aspects of network reachability.
- **Two-phase commit:** The data plane could also enable transactional configuration changes, such as a two-phase commit, to allow the decision logic to synchronize changes to the network. With good time synchronization (e.g., through NTP, or having a GPS receiver at each router or Point-of-Presence), the decision plane could instruct the routers to switch from one set of routes or packet-handling policies to another at a specific time, resulting in an infinitesimal convergence delay.
- **Fast failover:** The data plane could support immediate local reactions to unexpected network events, such as failures. For example, the data plane might have a table that indicates how to adapt packet forwarding after a particular link or path fails (e.g., the mechanisms defined for the MPLS fast reroute [1]). Local support obviates the need for the routers to contact the decision plane before reacting, while still allowing the decision plane to directly control the network by precomputing how the routers should react.

We believe the design of data plane mechanisms to support network-wide control is a promising new research direction.

2.2 Building on Existing Trends

We believe that our argument for moving the decision logic out of the routers is the natural extension of the centralized configuration of routers in large IP networks. Our proposal also takes other important trends to their logical conclusions:

- **Use of measurement data for running the network:** Measurement plays an increasingly important role in network management. Traffic measurements are used in detecting and diagnosing DoS attacks, tuning routing protocol parameters, and planning the outlay of new capacity. Routing protocol measurements are used in constructing a real-time view of the network topology, identifying sources of routing instability, and detecting and diagnosing anomalies such as black holes and forwarding loops. Data from the underlying transport network provides visibility into equipment failures (e.g., flaky optical amplifiers) and the mapping from the layer-1 and layer-2 facilities to the IP links. We believe that the dissemination of measurement data to the management systems should be recognized as a primary function of the network elements.
- **Open interfaces for router software and hardware:** Driven by the desire to separate router forwarding from protocols and network services, significant prior work attempted to define an open router interface analogous to OS interfaces at end-systems [2, 3, 4]. Recent standardization efforts within the IETF reflect this desire [5, 6]. Specifically, the IETF ForCES group [6] has proposed a framework that partitions a network element into separate control and forwarding elements, which can communicate over a variety of media. Whereas these efforts attempt to modularize the architecture and the functionality of *individual* routers, we propose to move most of the functions currently in the control plane features *out* of the routers altogether into a separate decision plane for the network.
- **Movement of path computation to separate servers:** Several recent proposals [7, 8, 9] argue for separating the computation of routes from the individual routers. These proposals aim to simplify network management and enable new features without modifying today's routers and routing protocols. We also argue for placing the key functionality outside of the network but go further in three respects. First, we believe that the management plane should configure the data plane plane *directly*, rather than driving the control plane by sending BGP or MPLS messages to routers. Second, we believe that the management plane should dictate other aspects of network operations *beyond routing* (e.g., packet filtering and quality of service). Third, we believe that the routers should have *much less control plane state and logic*, and perhaps ultimately have new features in the data plane to support the decision plane.

In addition, we argue that technology trends facilitate placing more functionality outside of the routers. We believe that cheaper memory and faster processors enable a small number of computers to store and manipulate the state quickly enough to drive the data plane for the routers in a large network.

3. Illustrative Examples

This section presents four examples that highlight the value of a network-wide view and the importance of a decision plane that can make coordinated decisions based on that view. The examples illustrate the need for joint optimization of multiple metrics, an integrated view of mechanisms affecting the same function, visibility across protocol layers, and efficient use of router resources. For each example, we highlight the challenges of solving the problem in the control or management plane, and illustrate the advantages of placing the functionality in a decision plane that has direct control over the routers in the network.

3.1 Joint Optimization of Multiple Metrics

Example problem – traffic engineering: Traffic engineering involves adapting the flow of packets through the network based on the prevailing traffic and performance objectives. In practice, a network has multiple (sometimes conflicting) objectives, such as minimizing the maximum utilized link and bounding the propagation delay between each pair of routers. Satisfying multiple goals as the network and traffic conditions change is very challenging.

Control-plane solutions: Early attempts to engineer the flow of traffic involved extending the routing protocols to compute load-sensitive routes in a distributed fashion [10]. In these protocols, the cost of each link is computed as some (possibly smoothed) function of delay or utilization, in order to steer packets away from heavily-loaded links. However, routing oscillations and packet loss proved difficult to avoid, since routers were computing routes based on out-of-date information that changed rapidly, and the effort was eventually abandoned. To improve stability, the distributed algorithms were extended to compute a path for *groups* of related packets (e.g., IP flows) traversing a MPLS Label Switched Path. These load-sensitive routing protocols can have stability problems as well, unless the dynamic routing decisions are limited to aggregated or long-lived flows. Perhaps more importantly, the protocols require underlying support for signaling and distributed algorithms for optimizing paths for multiple metrics.

Management-plane solutions: Many existing IP networks have instead adopted a centralized approach for engineering the flow of traffic using traditional IP routing protocols (e.g., OSPF) [11]. In this scheme, the management plane collects measurement data to construct a network-wide view of the offered traffic and the network topology. Since the optimization of the OSPF weights is a NP-complete problem, the management plane conducts a local search through candidate settings of the link weights, looking for a solution that satisfies the various performance objectives. Considering additional performance metrics is as simple as changing the objective function used to evaluate the solutions. However, this approach has its limitations in satisfying different metrics for traffic to different destinations, and for avoiding disruptions during failures and planned maintenance. Ultimately, having a single integer weight on each link is not sufficiently expressive, though this approach has proven very useful in practice.

Decision-plane solutions: Tuning the OSPF link weights is an *indirect* way of solving the traffic engineering problem, forced by the design of the existing routing protocols. Instead, we propose that the decision plane solve the problem *directly*, based on a network-wide view of the traffic and topology provided by the dissemination plane. In particular, the management plane can compute routing solutions (e.g., a forwarding graph for the network) that satisfy the many performance objectives, and explicitly send the forwarding entries to the individual routers. We believe that this would reduce the computational complexity of the optimization problem and permit solutions that today's OSPF routing cannot achieve.

	Destination	A	B	C
Source	A		All	To C:80
	B	All		None
	C	From C:80	None	

Figure 1: Sample reachability matrix for hosts A, B, & C.

3.2 Integrated View of Related Mechanisms

Example problem – control of reachability: An important facet of network design is the implementation of a reachability matrix, which defines whether traffic originating from a given IP address and port number should be delivered to another given IP address and port number. Particularly for enterprise or government networks, the reachability matrix can be quite elaborate in order to compartmentalize the flow of information as typically required by a large organization. Figure 1 contains a sample reachability matrix for three hosts, A, B, and C. The matrix specifies that hosts A and B can communicate with each other without restriction; host C cannot communicate with host B; and host A can only send traffic to host C if it is addressed to port 80. Operators need effective ways to configure the routers based on the desired reachability matrix, and to verify that an existing configuration adheres to the reachability matrix [12].

Management-plane solution: Today's IP networks offer two distinct tools to use in implementing a reachability matrix: packet filters (configured in the data plane) and routing policy (configured in the control plane). The state of the art is to implement full (or partial) reachability with routing, and then to further restrict that reachability by adding packet filters. However, the management plane does not necessarily know exactly what forwarding tables the routing protocols would generate, without modeling the complex logic for selecting paths and constructing the FIB (e.g., [13]). In addition, the forwarding-table entries may be different after a link failure or a change in the routing updates sent by neighboring domains. This makes it extremely difficult for a network operators to answer the basic question "can host A reach host B while the network remains connected?"

Decision-plane solutions: The reachability problems becomes easier when the decision plane computes all of the state on behalf of the data plane. The decision plane can construct an appropriate combination of packet filters and forwarding table entries, and compute the transitive closure on each path through the network to determine which hosts can communicate. In addition, the decision plane can more easily limit the scope of routing information by declining to send a forwarding entry for certain destination prefixes to certain routers. This may limit the need to employ packet filters on routers in the interior of the network. Increasing the role of routing in *limiting* reachability (by omitting certain entries from the forwarding table) is appealing, given that packet filtering is an expensive operation in the data plane.

3.3 Visibility Across Layers

Example problem – protection in IP/optical networks: The performance and reliability of an IP network depends on the events in lower layers, such as the layer-2 (e.g., ATM and Frame Relay) and layer-1 (e.g., optical cross-connect) networks. As a result, multi-path routing and multi-homing of customers at the IP layer does not necessarily imply independent failure modes, since multiple links may share common facilities (e.g., an optical amplifier) at lower layers. To prepare for planned maintenance and possible failures, the network routing should account for the shared risks to prevent excessive congestion when the links are down.

Control-plane solutions: Addressing this problem in the control plane requires extensions to expose the shared risk links groups (SRLGs) [14] to the IP layer. In some cases, this requires extending the layer-2 or layer-1 protocols to provide this information in a standard form [1]. In addition, the distributed path computation algorithm in the IP control plane must account for the SRLG information in selecting routes. This leads to significant complexity in the IP control plane.

Management-plane solutions: The mapping of the layer-1 and layer-2 equipment to the IP links may be stored in an inventory database, populated automatically from the lower-level protocols or entered manually as part of the provisioning process. In today's IP networks, the management plane can consider the SRLG information in searching for a good setting of the IGP link weights while preparing for the failure or maintenance of the shared equipment. However, as in our discussion of traffic engineering, the management plane is forced to express its routing decision indirectly in terms of the existing control plane parameters (e.g., OSPF weights).

Decision-plane solutions: The decision plane would have access to the SRLGs and the mapping to the layer-1 and layer-2 equipment. Armed with this information, the decision plane can compute a new forwarding graph in advance of planned maintenance on the equipment, and direct the routers to use these specific forwarding entries. The decision plane could even plan the steps in transitioning from the initial forwarding graph to the new one, to prevent transient packet losses, in advance of disabling the equipment.

3.4 Efficient Use of Router Resources

Example problem — limiting routing-protocol state: The overhead of exchanging routing protocol messages and storing the dynamic state grows with the number of routers and destinations. Yet, networks have a mix of routers, some with more memory and processing resources than others. Building a large network requires care to ensure that no router is exposed to greater demands than it can handle.

Management-plane solutions: Today's control plane introduces a difficult trade-off between minimizing the overhead on the routers and limiting the complexity of the routing design. The problem is exacerbated by the fact that the techniques for improving scalability limit the visibility a router has about the rest of the network. Scaling techniques often have side-effects that influence the forwarding of data traffic, making it difficult for operators to reason about their designs and sometimes leading to routing anomalies such as forwarding loops and protocol oscillations.

For example, we typically think of link-state protocols as giving each router a complete view of the network topology. However, large networks often divide the routers into *areas* (in OSPF) or *domains* (in IS-IS), where the details of the topology and subnets are summarized. On the positive side, the routers require less memory and often do not need to react to link changes in remote areas. However, areas also affect which paths the data packets follow through the network, requiring the operator to consider the influence of the area structure when configuring the network. Clever configuration of address summarization at area boundaries reduces the influence on path selection, but in general minimizing the effects is an NP-hard problem [15].

As a second example, large networks typically cannot handle the overhead of a *full mesh* iBGP configuration to distribute BGP routing information to all of the routers. Instead, some routers are configured as *route reflectors* that select a single best route for each destination prefix and advertise this route to their clients. Yet, a route reflector does not necessarily select the same best route that its clients would have with all of the information at their disposal. These inconsistencies can lead to protocol oscillation and persistent forwarding loops [16, 17]. Operators must replicate and place route reflectors with the goal of minimizing the likelihood of introducing these unfortunate side-effects. Even when these anomalies do not occur, route reflectors make it more difficult for operators to reason about how data traffic will flow through the network [13].

Decision-plane solutions: The decision plane constructs a complete view of the network topology and routing information, and then computes the FIB entries for each router. The decision plane can exploit opportunities to reduce the size of the FIB, such as collapsing two adjacent subnets into a larger supernet when they are mapped to the same outgoing link(s). With knowledge of the memory limitations on each router and visibility into the entire network, the decision plane can calculate the smallest FIBs that do not create a forwarding loop or place too much load on particular links. Finally, in contrast to today's iBGP architectures, the control overhead on individual routers does not grow with the size of the network, since each router need only exchange information with the decision plane rather than other routers.

4. Challenges

Our design shares many challenges with other proposals that advocate centralized decision making. Considering this, we have a rich set of solution approaches to borrow from. Centralization does not preclude replication. Schemes, such as those discussed in [7], for avoiding single point of failures are applicable here. However, our design has a defining characteristic—a wafer-thin control plane. It is necessary to examine a little more closely the associated impact on the scalability of the system and its ability to respond to failures.

Scalability: Two questions are most critical: (1) How much

[1] Automatically identifying shared risks is difficult (if not impossible) in some situations, such as when two fibers lie in the same trench or tunnel, or when two amplifiers use a common power supply.

network capacity is consumed carrying dissemination-plane data? (2) In the simplest implementation of a decision plane, where a single server serves as the sole decision maker, will the server be overwhelmed by the dissemination plane data and will it have the CPU resources needed to compute the FIBs for the network?

To estimate the capacity demands of the dissemination plane, let us assume there are 1,000 routers in the network, with about 10 interfaces each, and 100,000 destination subnets to which routes must be calculated. Feeding the decision plane with a view of the network topology will require handling roughly 10 Bytes/link or $10^3 \times 10^1 \times 10^1 = 10^5$ Bytes for the entire network. A complete update of the topology once a second requires only 1Mbps of network capacity (of course, link failures can be announced more rapidly). Choosing egress points for external destinations might require the decision plane to hear about 10^5 routes from 10^2 border routers, each having 10 peers. If each route requires 10 Bytes, when the decision plane first starts up it may need to receive 10^9 Bytes, which will take 1 second over a 10 GigE interface. After start up, it can receive incremental updates that are much smaller. Similarly, in the worst case when all the FIBs in the network start off empty, the decision plane will need to write 10^5 routes of 10 Bytes each to 10^3 routers for a total of 10^9 Bytes, again requiring 1 second over a 10 GigE.

Similarly, even centralized computation seems quite feasible. The key observation is that the decision plane does *not* need to perform the sum of the work done by all the routers. There is substantial overlap between the computations performed on different routers, which a centralized server will perform only once, and centralization of decision making may simplify the structure of the network design, eliminating some computations completely.

Responsiveness to failures. When a failure occurs, how long might it take for the network to respond? In the worst case of an unanticipated failure, a network run by a single centralized decision engine might need to wait a maximum propagation delay for the engine to learn of the failure, a delay while the new paths are computed, and a maximum propagation delay while the new FIB entries are sent out the routers. In comparison, reconvergence in a network today requires at least a maximum propagation delay for notice of the failure to spread across the network and a delay while each router along the way recomputes its routes. At worst, the centralized approach costs an extra maximum propagation delay, and this is likely offset by savings in the time taken for route computation. From observations in operational networks, reconvergence time is typically measured in seconds. Propagation delay, typically 40-100 ms, is insignificant by comparison.

However, the decision plane need not wait for a link failure to plan how to react to it. If the data plane supports local failover (e.g., MPLS fast reroute, equal cost multipath), the decision plane can precalculate responses to link failures and store them in the data plane, eliminating the decision plane as a bottleneck. Such preplanning is easier in a decision plane with a complete view of the network and likely failure modes than in the distributed protocols in today’s control plane.

5. Summary

From back-of-the-envelope calculations and design sketches, we believe that separating the functionality of network control and management into information *dissemination* and *decision* planes is practical. Reducing the control plane to a wafer-thin dissemination plane eases the creation of a network-wide view and allows the design of the decision plane to be driven by reliability and scalability requirements rather than the capabilities of individual routers. The primary advantage of this refactoring is that it provides a way for network operators to *directly express* their intention for how the network should operate. However, it also opens the door to future research on how to better predict the behavior of the network when individual mechanisms such as routing, traffic engineering, packet filters, etc. are composed together.

6. References

[1] M. Alicherry and R. Bhatia, “Pre-provisioning networks to support fast restoration with minimum over-build,” in *Proc. IEEE INFOCOM*, March 2004.

[2] G. Hjalmtysson, “The Pronto platform - a flexible toolkit for programming networks using a commodity operating system,” in *Proc. International Conference on Open Architectures and Network Programming (OPENARCH)*, March 2000.

[3] L. Peterson, Y. Gottlieb, M. Hibler, P. Tullmann, J. Lepreau, S. Schwab, H. Dandekar, A. Purtell, and J. Hartman, “A NodeOS interface for active networks,” *IEEE J. Selected Areas in Communications*, March 2001.

[4] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The Click modular router,” *ACM Trans. Computer Systems*, August 2000.

[5] A. Doria, F. Hellstrand, K. Sundell, and T. Worster, *General Switch Management Protocol (GSMP) V3*. Internet Engineering Task Force, 2002. RFC 3292.

[6] “Forwarding and Control Element Separation Charter.” http://www.ietf.org/html.charters/forces-charter.html.

[7] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, “The case for separating routing from routers,” in *Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture*, August 2004.

[8] O. Bonaventure, S. Uhlig, and B. Quoitin, “The case for more versatile BGP route reflectors,” July 2004. Internet Draft draft-bonaventure-bgp-route-reflectors-00.txt, work in progress.

[9] A. Farrel, J.-P. Vasseur, and J. Ash, “Path computation element (PCE) architecture.” Internet Draft draft-ash-pce-architecture-00.txt, September 2004.

[10] A. Khanna and J. Zinky, “The revised ARPANET routing metric,” in *Proc. ACM SIGCOMM*, pp. 45–56, 1988.

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

[12] G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford, “On static reachability analysis of IP networks,” in *Proc. IEEE INFOCOM*, March 2005.

[13] N. Feamster, J. Winick, and J. Rexford, “A model of BGP routing for network engineering,” in *Proc. ACM SIGMETRICS*, June 2004.

[14] B. Rajagopalan, J. Luciani, and D. Awduche, “IP over Optical Networks: A Framework,” March 2004. RFC 3717.

[15] R. Rastogi, Y. Breitbart, M. Garofalakis, and A. Kumar, “Optimal configuation of OSPF aggregates,” *IEEE/ACM Trans. Networking*, vol. 11, pp. 181–194, April 2003.

[16] T. G. Griffin and G. Wilfong, “On the correctness of IBGP configuration,” in *Proc. ACM SIGCOMM*, August 2002.

[17] A. Basu, A. Rasala, C.-H. L. Ong, F. B. Shepherd, and G. Wilfong, “Route oscillations in I-BGP with route reflection,” in *Proc. ACM SIGCOMM*, August 2002.