A LOGICALLY CENTRALIZED APPROACH FOR CONTROL AND MANAGEMENT OF LARGE COMPUTER NETWORKS

by

Hammad A. Iqbal

M.S.E. (Telecom. & Networking), University of Pennsylvania, 2002

B.E. (Electrical Eng.), N.E.D. University, 2001

Submitted to the Graduate Faculty of

the School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Hammad A. Iqbal

It was defended on

December 14, 2012

and approved by

Taieb Znati, PhD, Professor, University of Pittsburgh

David Tipper, PhD, Associate Professor, University of Pittsburgh

Prashant Krishnamurthy, PhD, Associate Professor, University of Pittsburgh

Rami Melhem, PhD, Professor, University of Pittsburgh

T. S. Eugene Ng, PhD, Associate Professor, Rice University

Dissertation Director: Taieb Znati, PhD, Professor, University of Pittsburgh

A LOGICALLY CENTRALIZED APPROACH FOR CONTROL AND MANAGEMENT OF LARGE COMPUTER NETWORKS

Hammad A. Iqbal, PhD

University of Pittsburgh, 2012

Management of large enterprise and Internet service provider networks is a complex, error-prone, and costly challenge. It is widely accepted that the key contributors to this complexity are the bundling of control and data forwarding in traditional routers and the use of fully distributed protocols for network control.

To address these limitations, the networking research community has been pursuing the vision of simplifying the functional role of a router to its primary task of packet forwarding. This enables centralizing network control at a decision plane where network-wide state can be maintained, and network control can be centrally and consistently enforced. However, scalability and fault-tolerance concerns with physical centralization motivate the need for a more flexible and customizable approach.

This dissertation is an attempt at bridging the gap between the extremes of distribution and centralization of network control. We present a logically centralized approach for the design of network decision plane that can be realized by using a set of physically distributed controllers in a network. This approach is aimed at giving network designers the ability to customize the level of control and management centralization according to the scalability, fault-tolerance, and responsiveness requirements of their networks.

Our thesis is that *logical centralization provides a robust, reliable, and efficient paradigm for the management of large networks* and we present several contributions to prove this thesis. For network planning, we describe techniques for optimizing the placement of network controllers and provide guidance on the physical design of logically centralized networks.

For network operation, algorithms for maintaining dynamic associations between decision plane and network devices are presented, along with a protocol that allows a set of network controllers to coordinate their decisions, and present a unified interface to the managed network devices. Furthermore, we study the trade-offs in decision plane application design and provide guidance on application state and logic distribution. Finally, we present results of extensive numerical and simulative analysis of the feasibility and performance of our approach. The results show that logical centralization can provide better scalability and fault-tolerance while maintaining performance similarity with traditional distributed approach.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

This thesis is the culmination of my graduate studies, especially the full-time graduate work I undertook during 2005-2010. The ideas that led to the direction of this thesis came from my discussions with my thesis advisor, Prof. Taieb Znati, in the summer of 2006 on the key architectural issues facing the Internet and limitations of the attempts at their resolution. Those conversations led to my realization that maintaining the scalable and fault-tolerant nature of Internet design should be a key design parameter for any future network management design, which also needs to be flexible enough to coexist with legacy infrastructure. It is my hope that this thesis will help in the eventual realization of this vision.

This work would not have been possible without the support and guidance of numerous of my advisors, teachers, colleagues, friends, and family. They have contributed in countless ways to my personal and intellectual journey. I am deeply indebted to them and hope that I will be able to pay this debt forward. It is difficult to name all those who have positively contributed to this journey but there are still a few who I would like to acknowledge individually.

I am very grateful to my thesis advisor, Taieb Znati, for his guidance, mentorship, and support. Learning from him has been a very illuminating experience and I am especially grateful for the intellectual freedom he gave me during these years. I am confident that this has greatly helped in developing my skills as an independent thinker and problem-solver.

I am also thankful to the members of my dissertation committee, David Tipper, Prashant Krishnamurthy, Rami Melhem, and T. S. Eugene Ng for their feedback, participation, and advice. All of them have positively influenced the contributions of this thesis and helped in improving my skills as a researcher in this area.

I would also like to thank other members of the Telecommunications and Networking program at University of Pittsburgh for their support and guidance. My sincere thanks and gratitude goes out to Richard Thompson for his mentorship and support over these years. Interactions with him have provided me with invaluable lessons in research and life, and have motivated me through the bleakest periods of my graduate life.

I started my graduate studies at University of Pennsylvania and I would like to acknowledge the members of Penn faculty who have positively contributed to my intellectual journey. I am thankful to Lori Rosenkopf, Saleem Kassam, Len Cimini, Dwight Jaggard, and Ernest Gilmont for their role in shaping my critical reasoning and research skills.

Colleagues in my research group at Pitt have been very helpful and I value their friendship and support. My thanks to Ihsan Qazi, Hui Ling, Anandha Gopalan, PJ Dillon, Octavio Herrera, Mohammad Aly, Mehmud Abliz, Mahmood Elhaddad, Carlos Caicedo, Kevin Huang, and Prathiba Menon for all the wonderful memories of Pittsburgh.

I am also thankful to my employer, MITRE Corporation, for providing financial support during the last two years, and to all of my colleagues at MITRE who have offered encouragement during this time.

Finally, I would like to express my earnest gratitude to the members of my family. The love, kindness and support they have provided me have made all this possible. Thanks to my parents, Iqbal and Akhtar, for their enduring support, encouragement, and sacrifice through the years, and to my sister, Hira, for her love. I am specially thankful to my wife, Uzma, for providing love, support, and understanding that made the graduate life bearable. She has also been a most diligent proof reader of all my papers and manuscripts and has always provided feedback that has improved the quality of my work. Lastly, thanks to our children, Haider and Mariam, for the cheerful joy and happiness that they bring to our lives.

1.0 INTRODUCTION

Internet's architectural foundation was laid down in the 1970s by a team of researchers working under the networking research program of Defense Advance Research Projects Agency (DARPA). The team had a clear set of requirements for the design of a new packet based computer network [1]. Those requirements, which were applicable at the time, shaped the Internet's future direction by influencing its very basic characteristics. However, as Internet increased in size and permanence through our society, the requirements that the Internet needs to meet have greatly evolved due to a variety of technical and social influences. Since the beginning, much research effort has gone into improving the Internet and making it more scalable, widely applicable, and aligned with different technical needs over time. As a result, many extensions and incremental additions have been made to the original Internet with the intent of adding new functionalities or fixing existing ones. This evolution of the Internet still continues.

However, there are concerns that incremental addition of new functionalities is sometimes unmatched with the original design philosophy and, as a result, Internet is becoming "a patchwork of technical embellishments" [2]. Incremental changes also increase the overall complexity of the Internet's architecture, as change to one component often results in unanticipated or undesirable interactions with other components. These interactions increase the fragility of the overall design and make it difficult for both designers and network practitioners to deal with the increasingly complex and over-constrained network state. Nevertheless, incremental changes have been certainly useful in Internet's adaptability and will continue to be used in its future evolution. Indeed, there can be little justification for changing the entire architecture for each new functional requirement.

Moreover, there are other issues—most notably in the area of network control and

management—that do not seem to be readily solvable without rethinking some assumptions in the original design. Most management issues stem from the fact that the Internet was not designed for the widespread global deployment that it is used for today. Internet which started with a few connected nodes in the 1970s currently serves over a billion users around the globe [3], with millions of network devices used to access and run the network. This exponential growth, coupled with the myriad incremental patches in Internet's control structure, have outpaced the network control and management tools and techniques available to the Internet operators and designers, leaving the present day Internet error prone and difficult to maintain [4].

Network management complexity manifests itself in several ways to the detriment of Internet's designers and operators. First, despite years of incremental research on making networks more manageable, current network management practice involves an inordinate amount of human involvement, with manual configuration of network devices being the dominant mode of operation [5]. Due to the inherently complex nature of network management operation, the skill set required for this task has become quite specialized, and anecdotal evidence suggests a shortage of network operators skilled at the level needed in today's complex enterprise networks. Furthermore, the reliance on human configurations increases the likelihood of errors and misconfigurations. They not only affect the original network but often adversely impact the global Internet connectivity through the effects of Internet's inter-domain routing. In a study conducted in 2002, Mahajan et al. [6] reported that errors and misconfigurations affect up to 1% of global BGP table entries. Inevitably, this management complexity translate into a very high cost for network operators [7]. Yet, despite its complexity, network management remains unable to satisfy the requirements of today's network operators; network operations such as traffic engineering remain very difficult using the current routing protocols [8].

The purpose of this dissertation is to investigate an architecture that can be used to address the management complexity in the Internet. Instead of incremental additions to Internet's control "knobs", we rethink the fundamental structure of network management and the physical placement of its control functionalities. We ask ourselves how different control functions can be distributed across network devices in a way that optimizes the balance

between design robustness and management simplicity. Our approach adopts the vision of separating network control functionality from the physical devices involved in data forwarding, as originally proposed by the 4D architecture for Internet's redesign [4]. By extending this vision of centralized decision-making and management to a *logically centralized* physical implementation, we aim at improving the reliability of centralized network management and providing a more robust and scalable design.

This work is especially relevant to large enterprise and Internet Service Provider (ISP) networks where reliable operation and effective control over network resources is currently a significant challenge. Successful implementation of logically centralized network management architecture holds the promise of drastically improved support for network management, lower network operation and management (O&M) costs, increased network reliability, and lower cost of networking devices.

1.1 CONTROL AND MANAGEMENT IN PRESENT-DAY INTERNET

Design of the way a network's distributed elements are controlled and managed is perhaps the most important aspect of an architecture's design. Effective control and management of networks is a ubiquitous challenge for network operators. This is especially true in the case of large and geographically dispersed networks, such as global enterprise networks and first and second tier ISPs, where it is important to efficiently manage the network resources across a large number of heterogeneous network devices while meeting strict constraints on network availability and reliability. Additional challenges in the control of such networks arise as the robustness, scalability and responsiveness of control functions are impacted by scale and geographical dispersion.

The difference between control and management is worth highlighting in the context of this discussion. Management usually refers to the process of using the set of directives given by a human network administrator to set or change the network behavior. Control, on the other hand, refers to the dynamic decision making inside the network to produce a desired network state while adapting to internal and external events and changes. In the following

sections, we discuss the fundamental problems in the design of today's Internet that have contributed to the prevalent complexity of network control and management.

1.1.1 Distribution: A Key Factor in Management Complexity

One of the main causes of the difficulty in managing today's networks is the difficulty in configuring and managing various distributed algorithms that collectively control the networks. Today's networks are controlled by a variety of distributed routing algorithms, each working independently to achieve some network-wide objective, while operating collectively on diverse physical network devices. This has created a situation where each network functionality (e.g. inter-domain and intra-domain routing) maintains a distinct state across many different physical devices and is governed by its own set of configuration rules and protocol logic. This distribution of control state and logic makes it extremely difficult to control the interactions between different protocols and algorithms. Consequently, the management of typical data networks requires extensive manual configuration of individual protocol parameters, leaving the networks fragile [9, 4, 10] and insecure [11].

The inherent complexity of managing the operation of different distributed algorithms over a wide range of heterogeneous devices make the task of network management difficult and error prone. Path computation in today's Internet is governed by a variety of distributed routing protocols, e.g. OSPF and BGP. The logic controlling the operation of these algorithm, along with the generated state, resides on a variety of switches and routers across the networks. Each of these routing protocols utilize their own network discovery mechanisms to learn about the network resources and use their path computation logic to compute routing paths. The routing logic, in each instance, is governed by individually configured policies that require extensive pre-configuration to maintain uniformity. Network connectivity is usually a product of disjoint operation of more than one routing protocols and requires careful management of their interactions and dependencies to ensure that the desired network state is achieved.

1.1.2 New Demands on Route Computation

The original design of the Internet utilized simple distributed algorithms for shortest-path computation. Since then network evolution has introduced many new features that interact, or otherwise have dependencies, with the process of route selection. However, route selection process operates independently from other network mechanisms, such as those involved in address translations and access control. The dependencies between these mechanisms need to be carefully controlled by network managers as they affect network's security, integrity, and connectivity. Any change in the state of one process does not automatically results in adaptation by the others, and management intervention is often required to ensure that the joint operation of these mechanisms reflects the desired network behavior.

In enterprise and ISP networks, where fine-grained control over the routing decisions is needed to meet service obligations and Quality of Service (QoS) requirements, desired behavior is induced indirectly through intricate configurations of individual routing protocols. This configuration usually requires careful selection of parameters that in turn affect the route selection process. This process, currently used for traffic management in Internet [8], makes the task of network management very difficult as it requires indirectly inducing desired behavior in dynamic protocol operation through static configurations. Furthermore any change in network, e.g. due to link or device failure, requires management intervention and a new set of protocol parameters.

1.1.3 Fusion of Control Logic and Forwarding Hardware

Network management is also constrained by the current model of bundling control logic and data forwarding in the same device. The control logic in modern routers, that includes routing protocols and other mechanisms necessary for the creation and maintenance of network state, resides in a complex management software. This monolithic software implements the operating system, governing the low level device operations, along with the higher level protocols that govern the distributed operation of router's control logic. The implementation of router software is not standardized, and as a result each router vendor implements and markets a different control software. Even within the products from the same vendor, evolution

in router hardware and addition of new features makes it difficult for network managers to fully understand the devices in their networks. The increasing complexity of router software is reflected in the raw size of conventional router software — IP routers contain approximately 5-10 million lines of code [12]. Incremental solutions to overcome this complexity, including the use of better management tools, has been ineffective as it is difficult to keep pace with the changes in various device operations and technical advances.

1.2 RE-THINKING INTERNET'S DESIGN

It can be argued, from the previous section's discussion, that the root cause of management problems in the Internet stem from the basic design choices of its original architecture. The problems in network control and management, and other issues especially in network security, have led many prominent researchers to argue for a re-design of Internet in-line with the present and foreseeable future requirements. Such a re-design will benefit from the experience gained during the past several decades of networking research that was not available to the early researchers.

However, there are major practical issues with the implementation of a re-designed Internet. Internet is used around the globe as the primary communication technology with the conveniences of the web, email, web-based multimedia, and social media. There is a also huge network infrastructure built using the present-day Internet technologies with devices that will be inflexible to any major change in the network design. Ideally, any new design of the Internet should consider the dependence of the user community on existing services and the infrastructure, with its huge capital expenditure, that supports these services.

This requirement of backward compatibility places a burden, not faced by the original Internet architects, that is usually impossible to mitigate without sacrificing design purity and simplicity. Due to this reason, we intend to use a *clean-slate* approach in our investigation, de-coupling our design from the issue of backward compatibility, and not constraining it by the features and modalities of the current design. This approach is also a feature of several recent proposals for Internet's re-design, that are reviewed in Chapter 2. We believe

this kind of fundamental research is imperative for the community's understanding of the network design process and pushing the frontiers of networking research.

Here, it is important to realize that basic research in network management proposed in our framework does not equate to an advocacy for uprooting the entire existing network architecture. We do not anticipate our framework to substitute the current design and infrastructure around the globe, as doing so will not be practical in the foreseeable future. Instead, the realization of our framework can take place along side the present design, with partial or full deployments only in those ASes where the benefits afforded by the new design outweigh the cost of transition. We believe this is possible because of the AS-centric approach of the framework, that does not require global changes to be utilized, and the inherent flexibility of the Internet design.

Internet's design allows immense flexibility in accommodating new paradigms and heterogeneity. This flexibility is demonstrated in the numerous changes that have been adopted in its evolution, with MPLS as one of major examples. It is important here to highlight the contrast between our proposal and the schemes which have proven difficult to deploy in Internet. QoS and IPv6 efforts continue to face considerable resistance as they impact the underlying transport foundation of Internet—the TCP/IP protocols. On the other hand, our proposed changes have minimal effect on TCP/IP as they only impact the control structure of the Internet where localized deployments can remain insulated as long as the inter-AS interface is not disrupted. We believe that an implementation of BGP would not be difficult at the decision plane and will provide the necessary interface with other autonomous systems.

1.2.1 Centralization of Network Control

Centralization of network control and logic provides an alternative and attractive approach to tackle the challenge of management complexity. The main motivation behind this approach is the reduction in complexity from decoupling control functionality from data forwarding devices and using centralized algorithms for network control instead of distributed implementations of the same. Centralization of network decision-making naturally allows simpler implementations and provides a single point of network interface for management and policy

specification. This can be a significant improvement in large network management where individual control of thousands of network devices is a very difficult and costly task.

At the level of route computation logic, we note that some algorithms such as Dijkstra's are inherently centralized— even though the implementation constraints on traditional routers lead to distributed protocols, e.g. OSPF. We can find inherently distributed algorithms such as Bellman-Ford algorithm utilized in distributed protocols such as RIP at the other end of spectrum from Dijkstra's. But there is presently no approach available to network practitioners that would allow centralized implementation of control logic at the protocol level.

However, centralization can come with a trade-off of network reliability and robustness, if implemented without explicitly considering these as design goals. One of the main design goals at the beginning of Internet was robustness to device failures and adversarial actions. This goal has reflected in the control distribution of Internet where failures can be automatically compensated by distributed decision-making. On the other hand, a design where the entire control state and logic is centralized at a single place in the network — a physically centralized design — carries the risk of introducing a single point of failure. In order for the Internet to preserve its fault-tolerant character, these trade-offs between management simplicity of centralization and robustness of distributed control must be carefully considered.

This dissertation investigates the design of a logically centralized control and management plane that can efficiently reduce management complexity by providing the benefits of centralized control without the robustness concerns of a physically centralized design. We adopt and extend the design approach of network control centralization advocated by the 4D architecture [4]. The 4D architecture advocates a new layering design of the IP networks that separates the task of packet forwarding, a data layer function, from the task of network control, an operation and management function. This separation of data and control layers is in contrast with the current practice where the data forwarding mechanism and control logic are intertwined inside monolithic network devices, such as network routers or switches. This approach to network control necessitates the centralization of control state and logic inside a *logically centralized* Decision plane, that is responsible for collecting, computing,

and maintaining the state required by the network devices to operate.

The design of an efficient and robust Decision plane requires careful consideration of the decision plane efficiency and robustness. A physically centralized decision plane design was investigated in [13, 14] where replication of physical Decision Elements (DE) was used to ensure Decision plane robustness to DE failures. In this design, a DE collects the required network information, maintains the algorithms required for computing network state, and transmits this state information to the data forwarding devices. The fault tolerance of the decision plane is then augmented with multiple stand-by DEs which can takeover in case of failure. While such physical centralization is good as a first order evaluation example, practical deployment of a centralized network management architecture may be restricted by questions about the overall fault-tolerance, response time, and scalability of the physically centralized decision plane.

An alternative design approach is where the logical Decision plane is distributed over physically independent DEs. In this design, each DE controls a subset of the whole network, and works collaboratively with other DEs to achieve overall network control. We believe this addition of distributed structure is necessary to make the centralized management architecture scalable with the size of the network, as well as in making it more robust to DE failures. As an example, while a centralized DE design might be attractive for a small to mid-sized campus network, the network latency of a large geographically dispersed enterprise network would result in higher response times in case of failures, making such a choice unattractive. Also, we note that while the decision plane might be enrolled in traffic management, threat monitoring, and security tasks, the complexity of even the basic shortest-path reachability computation on a controller rises super-linearly with the size of the Autonomous System (AS) [13], indicating a maximum network size where such a design might be deployed. Complete centralization of control logic also invokes questions about the robustness of design in the face of failures; even more so since the distributed nature the Internet was a design choice to prevent its failure due to any localized event.

1.3 THESIS STATEMENT AND RESEARCH CHALLENGES

The central thesis of this dissertation is that *logical centralization provides a robust, reliable, and efficient paradigm for management of large networks.*

The main research questions that this thesis invokes, and this dissertation attempts to answer, are:

1. What guidance can we provide to the network operators about the physical design of their networks that will optimize network performance in a logically centralized architecture?
2. The present design of Internet places a strong emphasis on network continuity in the face of failures. How does the logically centralized decision plane handle failures and maintain the fault-tolerant character of Internet?
3. The decision plane needs to seamlessly handle events (e.g. device and link failures, device additions, etc.) happening anywhere in the network and provide a uniform interface to the data plane. What mechanisms would govern the decision plane operation?
4. Application design space for logically centralized decision planes offers much flexibility in choosing the placement of control logic and state information in the network. However, these design choices also possibly open up new and unexplored trade-offs to network practitioners. What factors and design trade-offs are present in application design for logically centralized networks?
5. Performance benchmarks of the present-day Internet protocols provide a natural basis for comparison against the performance of logically centralized networks. Is it possible (1) to identify metrics that would form the basis for comparison, and (2) evaluate the performance of logically centralized approach using models that closely resemble its intended deployment?

1.4 THESIS CONTRIBUTIONS

The contributions of this dissertation are enumerated as follows.

1. Our first contribution is the design for a *logically centralized and physically distributed* network control and management plane. We utilize the architectural vision of the 4D architecture to propose a clean-slate logically centralized decision plane, where network control and management functions are implemented over a set of physically distributed controllers (DEs). Our approach is aimed at exploring the design space between the extremes of purely distributed control and total physical centralization. We consider the trade-offs between the two design extremes considered in the existing research and propose an alternative that allows network designers and practitioners the ability to customize the level of centralization according to the requirements of their networks.
2. We investigate the problem of optimizing a logically centralized network's physical design from an operational perspective. Since the decision plane in logically centralized architecture will comprise of a number of DEs, the performance of the decision plane could be affected by the placement choice of DEs in the network. Optimization of the physical design will therefore be essential for the efficient operation of this architecture. The physical design includes the number, placement, and connectivity of DEs within an AS, as well as the connectivity between routers and DEs. We present a scheme to optimize the physical design for faster decision plane response time and lower convergence delays.
3. We present mechanisms that lead to a fault-tolerant design of logically centralized decision plane. As a first step in this design, we present an optimal algorithm to manage the associations between DEs and routers. This two-stage exact algorithm allows a decision plane to dynamically adapt to changes or failures at the data plane. This algorithm is then utilized in our DPP protocol for decision plane operation.
4. We investigate the trade-offs that exist in the design of decision plane applications for logically centralized networks. A case study based analysis of traffic engineering application design provides valuable insights into the design space and reveals the existence of three key design factors that need to be jointly optimized for efficient application design.

5. Finally, we present extensive evaluations of the proposed algorithms and techniques on large artificial and real-world topologies. Our evaluations shows that it is feasible to efficiently manage large networks using the logically centralized approach. Specifically, we found that our design was able to provide sub-second convergence delays to various network failures, which is on par with the reported best practices of optimized OSPF and IS-IS protocols operation used in traditional networks.

1.5 THESIS ORGANIZATION

Chapter 2 presents a brief background of the technologies that are most relevant to the understanding of this dissertation and surveys the existing research in network control and management. Chapter 3 introduces the Logically Centralized Decision Plane (LCDP) and presents schemes for the optimization of LCDP's physical design. Chapter 4 investigates fault-tolerance and robustness in LCDP networks, and presents the algorithm for the dynamic assignment of network devices to DEs along with a protocol for coordinated decision plane operation. Chapter 5 investigates the trade-offs in decision plane application design using traffic engineering as a case study. Results of the convergence performance evaluations of our techniques in the context of application design trade-offs are also presented in this chapter. Finally, Chapter 6 discusses future research directions and concludes this dissertation.

2.0 BACKGROUND AND RELATED WORK

This chapter presents technical background and an overview of existing research work related to this dissertation. A brief overview of some of the technologies that are related to this dissertation is presented at first. This technology background is followed by a discussion of related work in control and management in different types of networks. The chapter concludes with a discussion of the features that distinguish this dissertation from the related work.

2.1 NETWORK CONTROL AND MANAGEMENT IN INTERNET

This section presents a brief introduction of Internet's structure and a background of the technologies related to control and management of Internet.

2.1.1 Structure & Organization

The global Internet is a network that inter-connects millions of smaller networks together. These networks are, in turn, a collection of thousands (or more) of computing devices. Thus the global Internet is a *network of networks*, which provides a platform for connected computing devices to communicate with each other.

One of Internet's remarkable feature is the heterogeneity of the computing devices and networks that connect to form the Internet. The computing devices connected to the Internet range from large-scale static mainframes and supercomputers to much smaller mobile com-

puting and embedded devices. This heterogeneity is afforded by the core Internet protocols that provide a common medium for communication across the Internet.

The protocol used for communicating over Internet is *Internet Protocol* (IP). IP is universally used for providing the *Network Layer* functionality in Internet Protocol Suite [15], presented in Figure 2.1 [16]. IP provides a scheme for device addressing and specifies a format for data communication. Computing devices on Internet are assigned one or more IP addresses and communication takes place by sending IP packet(s) addressed to the destination's IP address.

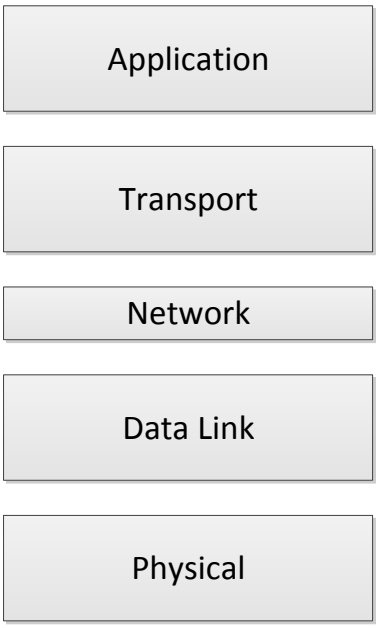


Figure 2.1: Five Layers of the Internet protocol suite

Structure of Internet is hierarchically organized. Users and end systems connect to Internet through enterprise networks or regional *Internet Service Providers* (ISPs). These local ISPs then usually connect with an access ISP at a higher tier level. The top-most tier is a collection of tier-1 ISP that have global coverage with high bandwidth links. These tier-1 ISPs are directly connected with each other and provide access to lower tier ISPs and large enterprise networks. Figure 2.2 depicts the hierarchical structure of Internet. Note that end-hosts and ISP customers can connect at any level of the ISP hierarchy. ISPs and

end-hosts can also be multi-homed, where they connect with multiple upstream ISPs for connection diversity.

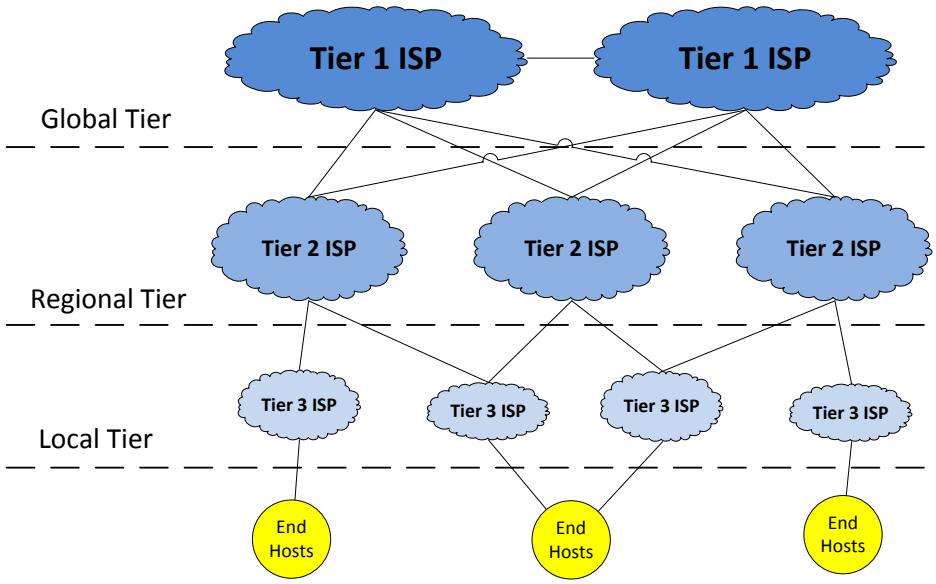


Figure 2.2: Conceptual structure of the global Internet

In networking terminology, an *Autonomous System* (AS) is a unit of routing policy in Internet. More specifically, AS is collection of connected IP routing prefixes that presents a common, clearly defined routing policy to the Internet [17]. An AS is generally synonymous with a single administrative ownership and control.

2.1.2 Routing & Control

The basic unit of communication over Internet is an IP packet; and *Packet Switching* is the foundational concept in Internet routing. Packet switched networks rely on the address contained in each packet for forwarding decisions. Each packet is also treated independently from any other packet. This is in contrast from *Circuit Switching*, commonly associated with telephone networks, where a unique communication path is setup before any communication takes place.

Packet switching requires a special type of network device dedicated to packet forwarding. This packet forwarding device is called a *router*. A router inspects destination IP address of an incoming packet and decides which outgoing link to use for forwarding based on the information contained in the router's *routing table*. Routing table is generally computed by a distributed routing protocol running on the same router, using the state configured by the network operator. Presently, every router needs to be configured in this way before it is able to participate in routing protocol exchanges and able to forward data packets. Router configuration commands are non-standardized and depend on the manufacturer and model of the router. A excerpt from a *config file* is shown in Figure 2.3 for a router running Cisco IOS. It is important to note, however, that router config files used in real-world routers often require thousands of line of policy configuration code [18].

Routing protocols specify how routers communicate with each other, what information exchange takes place in these communications, and how this information is translated into routing tables. In the present-day Internet, routing protocols are distributed in the sense that they are run independently on each participating router. Each router maintains its own protocol-specific state and uses this state to compute its desired routing table. The alternative approach of centralized route computation is not used as part of any routing protocols currently in wide spread use.

One of the two main kinds of routing protocols is *Interior Gateway Protocol* (IGP). IGP protocols are used to compute the routing tables inside an AS. The other kind of routing protocol is *Exterior Gateway Protocol* (EGP) that is concerned with inter-domain or inter-AS routing. This distinction between routing protocols is necessitated by the differences in policy and scalability requirements between intra- and inter-AS routing. Intra-AS routing has less stringent scalability requirements, as an autonomous system is considered limited in size. Moreover, since the entire AS is within the same administrative ownership, there are no restrictions on sharing AS network's detailed information between its routers. On the other hand, EGP routing needs to scale with the size of Internet and there are valid business and security concerns about the visibility of an AS's network internals outside its boundaries.

IGP routing uses two different techniques for its core state exchange and route computation operations. *Distance Vector* routing, used in RIP [19] and EIGRP [20] protocols, is

```
!
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.0
ip ospf 1 area 0
!
interface Ethernet0/1
ip address 192.168.2.1 255.255.255.0
ip ospf 1 area 1
!
ip router ospf 1
router-id 2.2.2.2
!
no bgp4 default unicast
bgp router-id 1.1.1.1
router bgp 40000
neighbor 10.0.0.1 remote-as 1
neighbor 10.0.0.6 remote-as 3
no synchronization
exit address-family
```

Figure 2.3: Excerpt of a conventional router configuration file

based on exchanging routing state among neighboring routers. As the neighbors exchange their routing state with their neighbors, updates and state changes are propagated throughout the AS network. On the other hand, *Link State* routing protocols, such as OSPF [21] and IS-IS [22], use network-wide flooding of state information. In these protocols, a router exchanges state information using Link State Packets (LSP) with all the routers in the network. There are trade-offs between the two routing approaches in terms of protocol message

overhead and convergence delay — with link state protocols offering better convergence at the cost of higher message overhead and protocol complexity.

2.1.3 Router Architecture

As discussed in the earlier sections, a router is a network device with a set of input and output interfaces. Its main function is to facilitate the forwarding of data packets by looking at their destination IP addresses, deciding the correct output interface using the information contained in the routing table, and transmitting them on the correct output interface. We categorize these functions as *data plane* functions as they relate to the task of data packet forwarding. Additionally, a router needs to compute its routing table using the distributed routing protocols discussed earlier. These route computation functions form the *control plane* of the router. Figure 2.4 depicts the logical separation between the two planes.

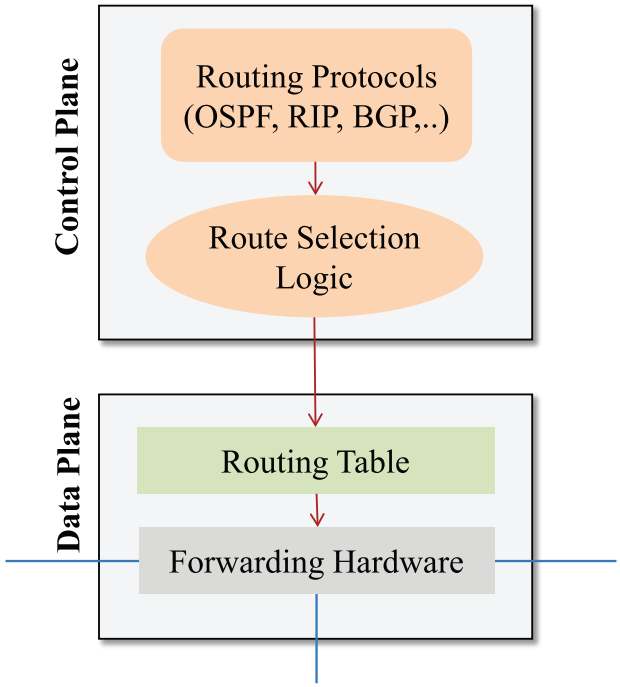


Figure 2.4: Conceptual design of a conventional Internet router

At the data plane level, a router may have several packets on its input interfaces destined to the same output interface. This means that there is often a need for temporarily storing

the contending packets and scheduling their transmission through the router's hardware fabric. *Input*-queued routers store packets at the input interfaces, *output*-queued ones store only at the output or outgoing interface, and *input-output* queued routers use a combination of both input and output queueing. Trade-offs in the selection of queueing mechanism mostly deal with the different hardware requirements of each mechanism. Most routers today use input-output queueing model [23].

From the preceding discussion, we can see that a router needs to perform several functions at the data plane in order to correctly forward a packet. A discussion of these functions follows.

Routing Table Lookup: An incoming packet at a router is most often forwarded based on its destination address, although a router can base the forwarding decision on other fields in the IP header in addition to the destination address. This decision on where to forward a packet is based on the information contained in a router's routing table, which contains a mapping between destination addresses and outgoing interfaces. Instead of listing each IP address possible, a routing table lists IP address prefixes and interfaces through which they are accessible. Therefore, each entry in routing table is made up of an IP address prefix and an outgoing interface number. The problem of routing table lookup is then to identify the longest prefix that matches the destination address of a packet, with a technique known as *longest prefix matching* [24].

Queue Management: This function is concerned with storing and managing packets within the limited buffers available in a router. Router buffers are limited in their capacity, especially in high-speed all-optical routers where buffers capacities are very limited [25]. Moreover, delay and jitter faced by packet traversing a network is dependent on router buffer sizes. Various techniques are available for deciding when, and which, packet needs to be dropped from an overpopulating queue [26].

Packet Scheduling: This function is concerned with selecting which packet to forward when an outgoing interface becomes idle. The most common packet scheduling mechanism in use today is *First-In-First-Out* (FIFO), where the oldest packet in a queue gets scheduled first. FIFO is an example of *work conserving* class of packet scheduling schemes that does not allow an interface to be idle as long as there are packets in the buffer destined to that

interface. Work conserving schemes are used virtually in all the routers in the Internet today.

2.1.4 Network Management Tools

The complexity of network management sparked many efforts to reduce the workload of network operators and make configurations less susceptible to errors. Most of these efforts lead to new tools and network protocols that are used for network management. However, as noted in the previous chapter, the underlying causes of network management complexity, i.e. distribution of control and fusion of data and control planes in routers, remain untackled.

At the protocol level, *Simple Network Management Protocol* (SNMP) [27, 28, 29] is the most widely used network management protocol. SNMP provides communication channels between a centralized *managing entity*, an application in network manager's workstation, and one or more *managed devices*, network devices that run SNMP protocol to expose their *Management Information Base*. SNMP facilitates network management by providing network operator with capabilities to "monitor, test, poll, configure, analyze, evaluate, and control" [30] the managed devices from a centralized location. SNMP does not, however, automate the task of network management by managing the network on its own; the burden of configuring and managing a network still remains on network operators.

There are several tools and proposals that aim at automating the task of configuration file generation. Most of this line of work utilizes existing network configurations, and similar config databases, to synthesize new configuration files. While these tools have been very helpful in reducing operators's workload, this approach has so far been limited to relatively simple configurations of new devices or validation of existing configs. The current state-of-art in network configuration management in the context of large ISP networks is discussed in [31]. Similar work has also focused on inter-domain routing configuration management [32]. There are also many commercial offerings in this area from router vendors [33, 34] and third party enterprises [35, 36].

There is also a substantial number of software tools designed to help the network managers in network visualization [37], data collection [38], traffic engineering [39, 40], and DoS mitigation [41]. The central problem these tools struggle with is they mostly assume certain

protocols and specific router software versions, limiting their general usability. Furthermore, they focus specifically on a subset of the overall network management problem. This also means that any interactions between different management mechanisms, e.g. between traffic engineering and packet filtering, are not covered.

2.2 EARLY RESEARCH IN NETWORK CONTROL AND MANAGEMENT

Network control and management issues have been explored in different networks from the very onset of networking research using different approaches. In this section we review the existing research work in network control and management. Different approaches used in Telephony, ATM, and other networks serve to highlight important design trade-offs and are presented in the context of TCP/IP networks, along with their relationship with the design choices used in our proposed framework.

2.2.1 Traditional Telephony Networks

The research community's experience in large-scale distributed network control and management started very early with the design of Public Switched Telephone Network (PSTN). PSTN's distributed control utilizes out-of-band signaling by the use of *Signaling System 7* (SS7) [42] — a packet switched control network logically separate from the managed telephone network [43]. SS7 is a message oriented distributed network which inter-connects network elements belonging to different administrative entities, and facilitates the signaling required for call-setup and management.

The out-of-band character of the SS7 system allows the control signaling to take place irrespective of the state of the managed network. This feature is in contrast with the in-band signaling found in the Internet where the control and data paths share the same links. The establishment of data paths require some control signaling to take place beforehand. However the control paths are themselves dependent on the operation of shared links, and any condition affecting the shared links such as link failure or congestion directly affects the

flow of control messages. The use of out-of-band signaling is an attractive alternative that can help reduce the complexity and potentially improve the performance of any clean-slate design.

2.2.2 Early Data Networks

Alternatives to the current IP network's distributed routing approach were explored early on by several specialized networks. Most notably among them were IBM SNA [44] and TYMNET [45]. Legacy IBM SNA employed dedicated network controllers to compute the routes in a session based host-terminal network. TYMNET used a single *Network Supervisor* to compute the routes for a virtual circuit based network. TYMNET's use of a centralized Network Supervisor is analogous to using a single Decision Element in a logically centralized control plane architecture. In TYMNET's case, the scalability of the network was constrained by the resource bottleneck at the Network Supervisor, limiting the network size to around 500 nodes. While realizing the technological advances in computation power and bandwidth availability, we believe that a physically centralized design would still be limited in a maximum network size because of the increase in the routing constraints required by various QoS, robustness, and security objectives, as opposed to the simple connectivity requirement in TYMNET.

2.2.3 Asynchronous Transfer Mode

The structure and characteristics of the SS7 networks formed the basis for the OSI model [46] for data networks, and are seen in the design of the *Asynchronous Transfer Mode* (ATM) networks [47]. ATM networks consist of three distinct planes: *User Plane*, which transports the user information along with the associated flow and error control information; *Control Plane*, which provides signaling for connection setup, supervision, and termination; and *Management Plane*, which co-ordinates among the different planes by providing fault, performance, configuration, accounting, and security management functions. The control and management issues in ATM networks differ significantly from the ones found in the Internet. The control in ATM networks pertains to the control of circuit-switched data flows,

and includes features that perform such tasks as admission control, virtual circuit setup, segmentation and re-assembly. These features and the use of circuit switching differentiates the scope of ATM's control and management from that of the Internet.

2.3 RESEARCH ON ROUTE COMPUTATION IN THE INTERNET

Control plane in the Internet generally refers to the distributed state and decision making of a number of routing protocols, e.g. RIP [48], OSPF [21], IS-IS [22], BGP [49]. This approach to network control is further affected by the presence of different middle-boxes [50] - devices that are placed in the path between the end-hosts and engage in activities other than routing without any communication with the routing protocols. Consequently, actual network control, or its routing design, comprises of the configuration of different distributed routing protocols and middleboxes that govern the network operation.

Complexity and difficulty of routing design, and the resulting configuration errors that affect network operation, is established by several research studies. Mahajan et al. [6] analyzed BGP route advertisements and found pervasive configuration errors reducing the efficiency of the routing design and affecting network connectivity. Usage of error-prone manual route configuration in enterprise networks and problems with automation were discussed by [51]. Maltz et al. [18] analyzed the configuration of operational enterprise networks and noted the large-scale of configuration settings and the absence of "interior" and "exterior" distinction in routing mechanisms used by network operators. Configuration error affect network components beyond the fundamental routing design, e.g. the impact of configuration errors on Domain Name System (DNS) was discussed in [52].

Problems in controlling and managing IP networks have led to the several attempts at alleviating the problem and making it easier to manage efficiently. Multi Protocol Label Switching (MPLS) [53] and its variants used a combination of semi-permanent resource reservation and explicit signaling for connection setup. MPLS helps network managers in provisioning and controlling aggregate traffic flows through their networks, and therefore serves as an essential tool for traffic engineering. However the problems in the underlying

network architecture remains unchanged. The centralized control over the establishment of paths afforded by MPLS is also reflected very prominently in RCP.

Routing Control Platform (RCP) [9, 54] was proposed as a logically centralized point for computing Border Gateway Protocol (BGP) routes and improving the scalability of large networks. RCP uses centralized servers for route computation and utilizes iBGP paths between BGP-speaking routers and servers. These design choices are very similar to the ones used in the 4D architecture. However, RCP is limited to BGP route computation and does not extend to the Interior Gateway Protocol (IGP) routes. Similarly, IRSCP [55] has been proposed as an intelligent route selector for a network, where it performs computation of BGP routes using not only the IGP information but also input from a intelligent system aware of other aspects of the network such as load conditions and DDoS attacks.

Another set of efforts for management complexity mitigation focused on simplifying and extending the router software design. SoftRouter architecture [56] advocated separation of control function from the data forwarding task of the routers, and provided protocols for binding between routers and servers implementing the routing protocols. Standardized signaling has also been researched as a way of enabling "programmable networking" in PRONTO [57], and design modularization is researched prominently by Click [58]. Similarly, Morpheus [59] provides an open routing platform for inter-AS routing. However, this line of research is constrained by their use of the Internet's distributed control algorithms, even in the case of centralized computation, e.g. in the "control elements" in SoftRouter's case. Therefore, although these proposals fix parts of the overall management problem, the root cause of management complexity remains.

Recent work on network management tools has also focused on network disruption minimization during outages, planned maintenance events, and major configuration changes. In this line of research VROOM [60] investigated migration of router state between physical routers using virtual machines. RouterFarm [61] focused on minimizing disruptions during customer re-homing at access routers. Migration of BGP sessions across routers was discussed in [62]. These proposals aim at improving the performance of current network management practices but do not tackle the core management complexity issues.

2.4 PROPOSALS FOR ARCHITECTURAL RE-DESIGN OF NETWORK MANAGEMENT

Several research studies realized the limitation of incremental solutions to network management problem and proposed designs which go beyond the backward compatibility constraint and involve some element of re-design of current networks. This line of research is closely related to the content of this thesis.

2.4.1 4D Architecture

4D architecture [4] decomposes a data network into four separate planes viz. Data, Discovery, Dissemination, and Decision planes. The data plane comprises of the routers, switches, and other network level devices. The main distinction between the 4D's data plane and the conventional network architecture is the lack of any control state or logic in 4D's data plane. Instead of using distributed protocols and requiring pre-configured state, the data plane devices in 4D architecture are governed by a centralized decision plane operating at the AS level. The decision plane is therefore responsible for collecting and maintaining information about the state of network devices and utilizing this centralized view for computing the mechanisms (such as routing tables) required by the data plane devices. As an example, the basic routing functionality can be implemented by collecting network topology, through the use of discovery & dissemination planes, and using it to generate the routing tables at the decision plane. These routing tables would be sent to the routers, using the paths established by dissemination plane, where they would be used for making packet forwarding decisions. Similarly, a decision plane is envisioned to control access (by configuring Access Control Lists), flow level authentication, traffic engineering, and other similar functions that can benefit from network-wide views, centralized decision-making, and direct control over network devices.

Figure 2.5 shows the layered design of the 4D architecture. This layering provides a separation of the data forwarding mechanisms, such as packet forwarding and filtering, from the state and logic required to manage the network. This separation is aimed at eliminating

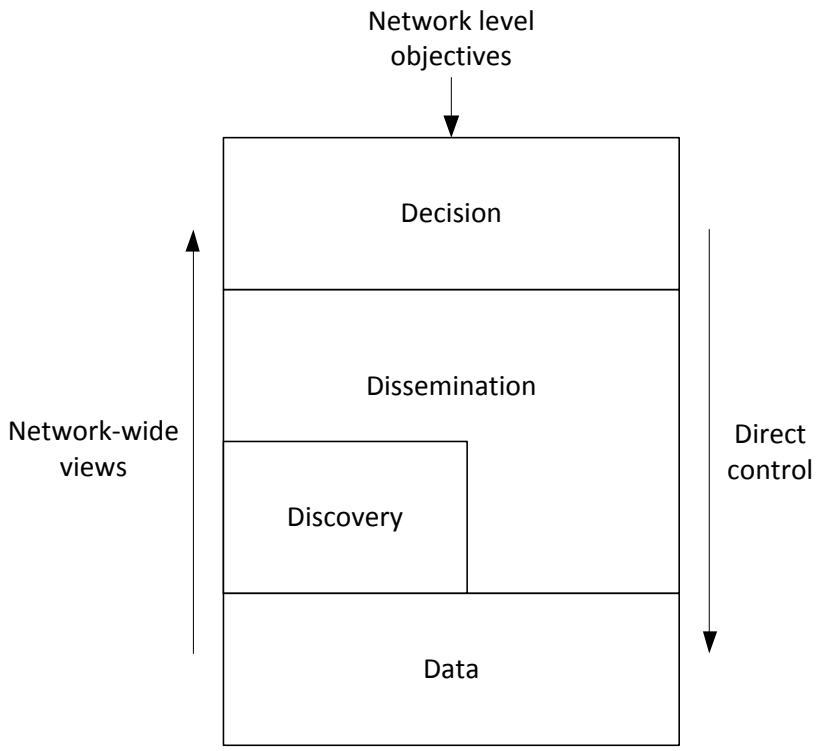


Figure 2.5: Layered design of 4D architecture [4]

the need for implementing complex distributed control at the device level of routers and switches, and moving the control functionality to the logically centralized decision plane.

The new decomposition of network functionality by 4D architecture eliminates the current management complexity of configuring myriad distributed algorithms and protocols at the device level. Instead of device level configurations, a network administrator using 4D architecture would define AS-wide policies that will be translated by the logically centralized decision plane into mechanisms and instructions needed by the network devices. The use of centralized algorithms at the decision plane, for tasks such as network-wide route computation, allow another opportunity for reducing complexity and misconfigurations as centralized algorithms require less configurations and often allow simpler implementations.

4D architecture [4] proposed a clean-slate approach to network control and management by decoupling the control logic from the data forwarding function of the routers and refactoring the control logic into a logically centralized Decision plane. The 4D Decision plane is not limited to route computation, and provides a single point for policy specification. Due to the network-wide visibility afforded by centralization of decision logic, the Decision plane is able to enforce the policies consistently over the entire network.

2.4.2 SANE

Secure Architecture for Networked Enterprise (SANE) [63] is a network architecture proposal focused around the concept of centralized network security control in enterprise networks. SANE provides a protection plane in the network that serves as the single point for access and routing decisions for all flows in the network. High level policy declarations form the basis for the protection plane's per-flow decisions.

SANE architecture is build on the observations that network security is critical in enterprise networks and the current approach of managing distributed routing and access control processes does not provide an efficient or robust solution. In addition to the complexity of managing myriad distributed protocols, the present architecture allows permissive connectivity at the link layer, requires that network devices trust multiple infrastructure components, and makes network information easily compromisable to an adversary. Thus by centralizing route management and access control, an enterprise network using SANE would need the centralized Domain Controller to explicitly allow each flow in the network and control the route over which the flow will traverse. SANE adopts several concepts of the 4D architecture: the clean-slate centralized approach to network management is apparent in the design of centralized Domain Controllers for joint computation of routing and access decisions. Furthermore, the use of source routed channel for management traffic, the protection layer, is similar to 4D's discovery and dissemination planes.

SANE presents a fault-tolerance scheme based on the replication of the centralized Domain Controllers. In this scheme, switches maintain routes to all of the Domain Controllers and randomly connect to any one of them for load balancing. This scheme implicitly assumes

the model of a data-center like localized network environment where the state across multiple controllers can be synchronized and the cost of querying any of the controllers is similar. This is a major difference between SANE and our work that, as discussed in Chapter 3, focuses on maintaining enterprise-wide control over large enterprise networks.

2.4.3 Ethane and NOX

A centralized network architecture for defining and enforcing fine-grained network-wide policies was described by Ethane [7] which builds on the SANE design. Like SANE and its predecessor 4D, Ethane uses centralized decision making and decouples control functions from data forwarding devices. Ethane targets enterprise network management and ensures the implementation of explicit routing policies declared over named network devices. The main difference between Ethane and SANE is that, unlike SANE, Ethane is compatible with IP protocol and does not require any changes to the end-hosts. Furthermore, Ethane switches are able to coexist with Ethernet switches–allowing incremental deployment of Ethane in a network.

Ethane uses a centralized design where a *Controller* is responsible for authenticating all network devices and flow establishment requests, computing paths for all flows based on specified policies, and , keeping the required state (network topology, registrations, and bindings between names and different address spaces) for the network. The controller is responsible for setting up each flow in the network. Each flow setup requires it to perform several sequential operations of varying complexity. In addition to the single point of failure that is possible, this design also raises concerns about the scalability of the controller as a centralized server could fail to scale with the number of flow requests in a large and dynamic network with significant end-host churn and/or mobility. Ethane's authors briefly discuss three approaches to address these concerns, two of which only improve fault-tolerance of the design. In the simplest two approaches, secondary Controllers (in cold or hot-standby modes) are added to serve as failover controllers. The downside of the first approach of cold-standby is that the secondary controllers don't have the replica of primary's state before it failed. Therefore, the new primary controller will need to recompute the entire network state, which

includes re-registering and re-binding all hosts, switches, and users. The second approach of warm standby controllers promises faster convergence as the warm standby controllers would have some, if not all, of the primary's state. The third approach of full-replication envisions multiple active controllers with weakly synchronized state. This approach is the closest to the logically centralized and physically distributed design discussed in Chapter 3. As our design does not assume any particular decision plane configuration, the mechanisms in our proposed framework that deal with the physical distribution of network-wide decision making are broadly applicable to Ethane as well.

NOX [64] follows the design of SANE and Ethane and proposes a modular control plane framework where management applications can access network-wide views. NOX's design does not require any per-flow coordination between controllers and only provides synchronization of network topology at the control plane. This limitation would present a problem for management applications that may need awareness of global flow states, for example minimum delay routing.

2.4.4 CONMan

CONMan [65] utilized the concept of management plane and centralization in the design and operation of "network managers" that are used to manage the protocols running on individual routers. The management plane in CONMAN, in similarity to the 4D Decision plane, is self bootstrapping and does not depend on the operation of data plane.

CONman focuses on reducing the complexity of configuring traditional routing protocols by exposing the minimal set of protocol-specific information that is required for the proper configuration of each protocol. In CONman, modules are network-wide objectives such as routing and their basic characteristics are called the Module Abstraction. All protocol modules in CONMan self-describe themselves using this abstraction. A protocol is modeled as a node with connections to other nodes, with its switching and filtering capabilities, performance and security characteristics, and certain dependencies on other protocols and processes. The centralized network managers in CONman collects the network state by soliciting the list of modules from each of the managed device along with its local physical

topology and module abstractions. Thus a network manager is able to collect network-wide state that it utilizes to translate high-level policies given by a network administrator to the generic mechanisms that are needed to satisfy the policies. These are communicated to the devices using the management channel between network managers and data plane, which borrows the 4D dissemination and discovery plane concepts.

CONman's management plane is limited to providing an interface for communication with the routers, where the actual control functionality resides. Therefore, while CONMAN provides a solution to configuration management, the underlying complexity of distributed routing algorithms remains.

2.4.5 Maestro

Maestro [66, 67] provides an "operating system" for the network control applications, that are implemented in a modular fashion, and handles their concurrent operation. The basic premise of Maestro is that network management functions have interdependencies that need to be explicitly managed by a centralized entity. The Maestro design provides the management platform for the network management functions that are implemented as modules on top of abstraction layer provided by it. The functions provided by the Maestro's management layer are communication, scheduling, feedback, concurrency, and transition. Different modules implementing network control functions, such as intra-domain routing, will utilize Maestro's management abstraction layer to jointly drive the state needed for the network. Maestro's network operating system functionality is conceptually similar to NOX [64] and other similar current efforts in OpenFlow controller development [68].

Maestro's design builds on the concepts of 4D architecture and provides a framework for a centralized and modular decision plane. In that respect, Maestro is complimentary to our work and the concept of modular operation of different decision plane functions can be extended from a physically centralized implementation to one where the network operating system is virtualized over a set of servers that jointly control the network state.

2.4.6 ONIX

ONIX [69] builds on the work of Ethane and provides a general framework for the implementation of distributed OpenFlow controllers. Conceptually, Onix internally maintains the network state information in its data model and provides programmatic access to the control layer for this network data. The control logic is not defined by Onix; it is also expected that the control plane logic will provide mechanisms for checking and maintaining data consistency between different instances of Onix as well as between Onix and the network elements. Scalability of Onix-based networks is supported by partitioning, where network may be logically partitioned to report to different Onix instances, and aggregation with hierarchical structuring of Onix controllers.

Although Onix provides low level mechanisms that provide building blocks for the control logic, it does not aim to spell out their design. The methods for managing and recovering from failures is an example of a function that control logic will need to provide. On the other hand, this thesis is more devoted to the investigation of how the control logic/plane will be architected. The platform provided by Onix or other similar designs can be leveraged by the control plane design provided in this work.

2.4.7 HyperFlow

HyperFlow [70] presents a brief vision for a logically centralized OpenFlow control plane that is based on a distributed file system for state synchronization. This paper recognizes the scalability limitation presented by most of the existing single controller approaches in the literature and argues for a design that allows local control of switches, synchronization of network-views, and resiliency to network partitions.

HyperFlow's design is based on publish-subscribe paradigm, where each controller selectively publishes events that are related to network state changes, and other controllers replay the events to construct the overall network state. A distributed file system with build-in guarantees for event storage, ordering, and partitioning resiliency provides the underlying data management functions that are utilized by HyperFlow. Each controller in HyperFlow is assumed to have consistent network-wide state which is used to execute the same software

and applications on the entire set of network state. This means that even though controllers would be managing different set of switches, they are expected to run "as if they are controlling the whole network". This assumption of strict synchronization of state and logic is in contrast with the weak state synchronization guarantees that are possible with HyperFlow's dependence on distributed file system.

The overall goals of HyperFlow are similar to the ones developed in this thesis. The concept of logically centralized and physically distributed control plane, that was developed in the earlier investigations [71] of this thesis's core contributions, is the basic design goal of HyperFlow. However, the mechanisms used for achieving logical centralization are different along with the scope of the work. Whereas HyperFlow targets OpenFlow controllers (in particular, NOX [64]), our work presents a more general framework which can be adopted for distributed control plane implementations, irrespective of the control plane applications. Furthermore, we present a finer grained design with an emphasis on large enterprise network design with discussions on data-control plane associations, level and scope of control plane state replication, and dynamic control plane failure recovery.

2.4.8 Other Notable Proposals for Internet Re-design

Active networks [72] describe a way to add customized router-based computation and state in the network. In an active network, the routers or switches of the network perform customized computations on the messages flowing through them. For example, a user of an active network could send a customized compression program to a node within the network (e.g., a router) and request that the node execute that program when processing their packets. This approach can be extended to cover network control functionalities in a way that gives more control to the end-points over the network. Active network design bring several new questions to the architecture research. One of the most important challenge in allowing user originated code to be executed in network devices is the issue of safety - stopping a malicious user or a misconfigured machine from injected harmful code. Similar in its end-goal of facilitating the deployment of new services, NetServ [73] proposes installing virtualized service modules on the router control plane. Active networks' approach of maintaining a baseline set of

mechanisms in the routers that can be controlled remotely is shared in its broad concept by the 4D architecture.

Role-based architecture [74] proposes the decomposition of layering design into much smaller "role" abstractions. It replaces the rigid design of stacked protocol layers by a low-level heap of protocol units, allowing all the network components to be explicitly identified, addressed, and communicated with. This architecture uses meta-data in packet headers as an in-band signalling mechanism for communication with roles defined in the middle-boxes.

Both of these proposals were motivated by the need to reduce complexity in Internet control plane. Although they differ in their solution approach in key respects with this thesis, there is similar recognition that the complexity and rigidity of the current control and management design is undesirable for the future growth of the Internet.

Autonomic Network Management architecture [75, 76] presents a vision for networks made up of self-configuring, self-organizing, self-federating, and self-healing nodes. This architecture envisions a system which as whole attains a higher degree of automation than simply the sum of its self-managed parts [75]. The ANA architecture's key concepts go beyond the network management aspects that this thesis is focused on, and are generally not yet grounded in practical details. However, we note that the vision for self-configuring networks is similar to 4D and our design's emphasis on the decision plane's ability to configure newly connected nodes without requiring extensive a priori manual configuration.

2.5 SUMMARY

This chapter presented background information on several foundational concepts that are leveraged in this thesis. The complexity of the current control plane mechanisms, with its dependence on several distributed routing protocols, was highlighted in contrast with the relatively simple primary task of a router—packet forwarding. This complexity, along with the prevalence of manual configuration, is affecting the stability, efficiency, and extensibility of the Internet's architecture.

The primary focus of this chapter was on the closely related research proposals on In-

ternet's clean-slate redesign that were also focused on reducing Internet's management complexity. 4D architecture was summarized here along with the other proposals that looked at similar problems. This review suggests that although there is much activity on the vision of a centralized control/decision plane as a way of mitigating network management complexity, there has been limited effort on the design of this control plane that is also efficient and scalable. Most of the existing work is based on a physically centralized design as the first-cut approximation of the re-designed control plane, and there is a need to look at logical centralization and its associated trade-offs as a way of improving design scalability and performance.

This concept of logically centralized and physically distributed control plane is presented in the next chapter and the rest of the thesis is devoted to further development of the mechanisms that can help in the realization of this concept.

3.0 FRAMEWORK FOR LOGICAL CENTRALIZATION OF NETWORK MANAGEMENT

This chapter introduces the concept of a *Logically Centralized (and physically distributed) Decision Plane*(LCDP), where the traditionally distributed functionality of route control and management is replaced by logical centralization of network views and control logic in a decision plane. The logically centralized decision plane need not be a single physical device – in fact it is desirable to have redundancy at decision plane for fault tolerant and scalable network architecture. The LCDP framework provides for a decision plane design that is distributed over a physical set of Decision Elements (DE). These DEs collaborate to present a unified view of decision plane and provide network-wide control and management service to the rest of the autonomous system network. In the next section, network model of our framework is presented along with the underlying assumptions, followed by the details of LCDP and rationale of its fundamental design choices. The following section discusses how LCDP model can be implemented in network, from a practical standpoint, and provides a scheme for the optimization of DE placement in a network. The DE placement optimization algorithms are presented next along with the results of their evaluations on ISP topologies.

3.1 NETWORK MODEL

This section discusses the network model that is considered in this dissertation, and the design and assumptions that underly the architectural choices in LCDP.

The primary network model considered in this thesis is a relatively large sized network that is under single administrative control. The size of the network considered here can

be along the dimensions of number of network infrastructure devices, the geographic scale of the network, and the size and complexity of the independent control and management processes that govern route management of today's networks. Typically large enterprise and Internet Service Provider (ISP) networks fit this network profile. The infrastructure devices considered here are primarily the routers and switches of the network, but the LCDP design offers flexibility to manage other network devices that have a dependency with route management process even though they don't participate in route computation in traditional networks. Firewalls offer the chief example of such devices which could include network address translators, load balancers, and WAN accelerators. The chief reason for choosing this network model is that the network complexity and the corresponding Operations and Management (O&M) costs tend to be highest in such networks.

A transition to LCDP network management paradigm will likely provide the highest return on investment for this network profile by reducing the high O&M costs. However, this will need to be offset by the cost of transitioning to the LCDP model. If the transition cost is linear function of the number of devices, the total transition costs could be a negative factor for an enterprise with a large network that is considering LCDP architecture. However, we posit that the transition to LCDP architecture could be gradual process where traditional network devices could be made to work alongside (or within) LCDP and replaced as part of regular lifecycle refreshes. We assume that given the O&M differences between centralized management in LCDP and the distributed route management in such cases, the cost of transitioning to LCDP architecture would be overshadowed by the savings in O&M costs, making the transition an attractive choice for the network profile considered here.

Similar to the network device count, the geographic scale of a network also generally corresponds to higher O&M costs. Management of a geographically dispersed network usually translates to careful orchestration of several independent networking technologies. In addition to Internal Gateway Protocol routing (IGP) that is a common denominator in networks of all sizes, these networks have to deal with inter-domain routing, path management through tunneling or MPLS, iBGP meshes, DNS management, and other technologies which are either typical of geographical dispersed networks or whose complexity increases faster with geographical dispersion. The corresponding O&M cost of managing these technolo-

gies is even further increased by the limits placed by the inherent nature of geographically dispersed operations. Even when the administration of a large geographically dispersed network is centralized there is generally a need to retain site personnel to locally administer the devices in their region in case the path to centralized administrator fails, taking down the in-band management path used in traditional routers.

3.2 LOGICALLY CENTRALIZED DECISION PLANE

This section presents the vision of a *logically centralized* network management entity, a decision plane, that is *physically distributed* over a set of network management servers, referred here as Decision Elements or DEs. The distributed DEs cooperatively manage the network by assembling a complete network-wide view of relevant network information (such as topologies, traffic matrices, device status and parameters) and using that to translate high-level network objectives into mechanisms that are used to exert direct control over the operation of networking devices.

3.2.1 Logical Centralization

The sub-optimality in the current distributed design is inherent in the mechanisms that allow the operation of routing protocols to scale enough to meet the current reachability needs. Even without considering the case of inter-domain routing, which is beyond the scope of this thesis, there is usually significant routing sub-optimality [77] in intra-domain routing design of the current networks. There are several causes of this routing sub-optimality and they can be traced to the need of supporting IGP routing protocols that can scale to larger AS sizes. This is an area in networking research that has received ample coverage over time [78, 79, 77]. At the same time, the problem is difficult to solve while remaining constrained by the fully distributed design of routing protocols. For example, the computational load on a router's CPU, offered by the distributed routing protocol processes running on it, provides a constraint that limits utilizing routing protocols that might provide better routing optimality

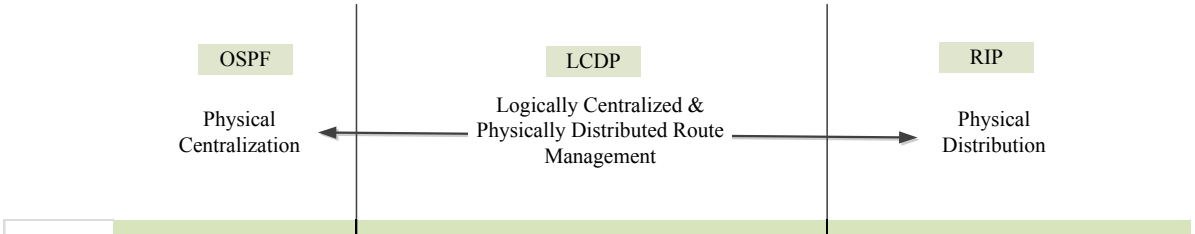
at the cost of additional CPU load. Furthermore, flooding of link-state advertisements by OSPF provides an efficient mechanism for the dissemination of routing information within an area, but at the expense of significant network bandwidth use that limits the scalability of the design. Logical centralization of route management could provide a means of overcome both of these resource constrains as centralization route computation will offload the routing load from router CPUs and, at the same time, directed dissemination of network information could remove the need to broadcast link state messages.

Centralization of network control logic and state becomes even more important when we consider the additional demands that are placed on the router's limited computational and memory resources by needs that go beyond simple network reachability. Traffic engineering is an example of a well-defined need that is difficult to meet with today's distributed IGP routing [8].

The mechanisms that control the network operation in today's Internet are overwhelmingly either fully centralized or fully distributed. This has created an inflexible design where network operators have limited ability to customize the network management mechanisms according to the needs of their networks. Routing provides a good example of this problem where IGP routing is either fully distributed or fully centralized, at least in protocol operation, depending on the chosen protocol. In the case of link-state routing protocols, such as IS-IS and OSPF, full network state is collected by each router and shortest paths routing algorithms implemented in each router's control plane compute the paths to every other router and destination. This operation is fully centralized as each router collects and computes paths on the full network topology. On the other end of the spectrum from fully centralized route computation, we find distance vector routing protocols, such as RIP, where both network state and route computation is distributed among the participating routers. Routers using RIP protocol only exchange route information with their neighbors and don't build the map of network adjacencies as needed by link state routing. This enables RIP to operate at a lower cost of protocol bandwidth overhead, as broadcast based message dissemination is not required.

As depicted in Figure 3.1, the logically centralized design aims at providing a balance between the two extremes of distribution and centralization. In this design, the primary

objective is to provide network operators with a customizable network management platform which they can configure to match the level of centralization needed in their networks. In LCDP, customization of the level of centralization is achieved by the location and number of DEs deployed in the network. This is in contrast with the link state approach, where each router's control plane works independently, and also with distance vector approach, where the routing protocol instances of the entire set of routers collaborate to jointly form the network control plane.



	OSPF	LCDP	RIP
Network State	Centralized at each router	Centralized at selected network nodes (DEs) where full network-wide visibility is maintained	Fully distributed among participating routers
Route Control Logic	Replicated at each router	Logically centralized at selected network nodes (DEs)	Replicated at each router
Mode of Information Exchange	Link state information is broadcasted by each router	Minimal information exchange between routers; unicast communication between routers and DEs	Control decisions are exchanged between nerighboring routers

Figure 3.1: Spectrum of control state and logic distribution in route management

The LCDP design utilizes the abstraction of 4D architecture [4], which decomposes a data network into four separate planes viz. Data, Discovery, Dissemination, and Decision planes. We embrace 4D architecture's concept of decision plane centralization with the realization that the distribution of the control and management functionality in traditional Internet design is sub-optimal—and unscalable, when additional requirements beyond simple reachability are considered.

Figure 3.2 illustrates the concept of logical decision plane centralization in an example ISP network that spans the United States with several Points of Presence (PoPs).

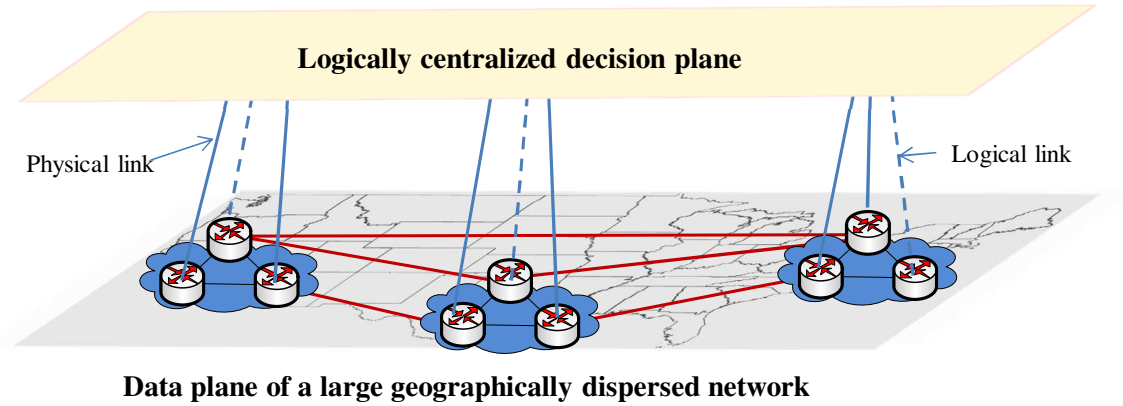


Figure 3.2: Logical centralization of decision plane

Figure 3.2 highlights several key design aspects of the proposed architecture. First, the logically centralized decision plane presents an AS-wide coordinated and unified view of decision plane functionality to the network devices that it manages. Therefore, from the perspective of a router or switch the decision plane—that may although be comprised of a set of discrete physical devices— appears as a single plane. A router or switch in this design will have very little, or no, configuration that directs its communication to any particular DE. A new or rebooted device needs only to reach the logical decision plane, through the functionality provided by 4D architecture's discovery plane, for it to be included in the network. Source routing paths, established as a result of regular beacon messages broadcasted by the DEs, is one implementation of 4D's discovery and dissemination plane that can be leveraged for this, as discussed by Greenberg et al. [13]. The computation of routing tables, or any other control state, should be seamlessly handled by decision plane from the establishment of path between the new network device and the decision plane. This, in turn, requires the decision plane to be able to handle failure among itself, i.e. within the set of DEs operating at the decision plane, and at the data plane level, i.e. in the paths between the decision plane and the network devices. Chapter 4 discusses the mechanisms that can allow LCDP to cope with failure.

Secondly, the paths shown in Figure 3.2 show both physical and logical connectivity

between the routers and DEs. The logical paths in the design traverse multiple physical links in the network's data plane to reach the decision plane. The establishment and maintenance of these paths is governed by the dissemination plane.

It is important to highlight that physical centralization of network control logic is undesirable in order to avoid potential problems with scalability and fault-tolerance. Logical centralization of the decision plane is preferred alternative that could be realized using a set of Decision Elements (DEs) which will collaborate to perform the function of network-wide route computation, adding a level of distribution in the decision plane. This concept of decision plane's physical distribution is discussed next.

3.2.2 Physical Distribution

The two main reasons for physical distribution of decision plane are increased scalability and fault tolerance of the design. Scalability is considered here as the ability for the decision plane to scale to AS networks of arbitrary large sizes. Existence of power-law function in AS sizes [80, 81] indicates that AS with very large sized networks will remain a feature of the Internet. Indeed, the motivation for adopting an architecture such as LCDP is greater for such large networks due to the potential for proportionally large O&M savings from centralized network management with reduced human involvement. Even if the large computation load offered by the route computation process in a large-sized AS can be handled by a physically centralized decision plane, the convergence behavior of that decision plane may not scale in a network that is geographically dispersed at the same time.

Propagation delays between the DE and the network devices, even when the location of DE is optimally chosen inside the network, are large enough in commonly seen geographically dispersed topologies [82] that any change in the network will result in large transient periods in the network, while the route computation process at the physically centralized DE is in process or the routing tables sent from the centralized DE have not been received by the far reaches of the network. These transient periods can contain routing loops as some routers will start forwarding packets based on forwarding tables received from the decision plane, while others that have not received the updated forwarding tables will continue to forward

packets based on that old state [83]. By distributing the route computation load over a set of physically distributed DEs, the scalability of the architecture is improved, given that routers are associated appropriately with DEs that are closer in terms of latency. This problem of router to DE association with the objective of minimizing the routing convergence time over the decision plane is considered in Chapter 4.

Physical distribution is also important for the fault-tolerance of the LCDP design. Here, we consider two reasons for this statement: first, from the perspective of network devices; and second, from the perspective of LCDP operations. From the network devices perspective, the LCDP design abstracts the decision plane functionality and decouples it from the physical location of route computation (and other decision plane functionality). It follows from this abstraction that unless there is a robust mechanism in place at the decision plane to failover the decision plane services provided to a router, there could be periods of time where local DE failures may disrupt the network-wide control of devices from LCDP, i.e. the router may be un-governed if the DE serving it fails. From the perspective of decision plane operations, a fault-tolerant decision plane design necessitates a distributed approach where DEs are placed along the geographical structure of the underlying AS network.

The fault-tolerance aspect of the LCDP design is depicted in Figure 3.3. Normal operations of a LCDP-managed network is shown in Figure 3.3a, where a couple of DE are jointly managing an ISP network with two PoPs with each DE managing the operations of one PoP. Here hardware redundancy, e.g. with active-standby DE design [13], could ensure that local hardware failures don't disrupt the decision plane operations. However, there are still plenty of failure modes common to the locally redundant DE design, and geographical diversity in the LCDP is leveraged for a more resilient design with mutual failover between DEs. This is shown in Figure 3.3b where DE2 takes over the management of PoP1 after DE1 fails. Algorithms for enabling DE failover are discussed in Chapter 4.

3.2.3 Local Area Policies

This section discusses the possibility of "local" policies for each area in the LCDP-managed AS, and the trade-offs in the degree of decision plane centralization.

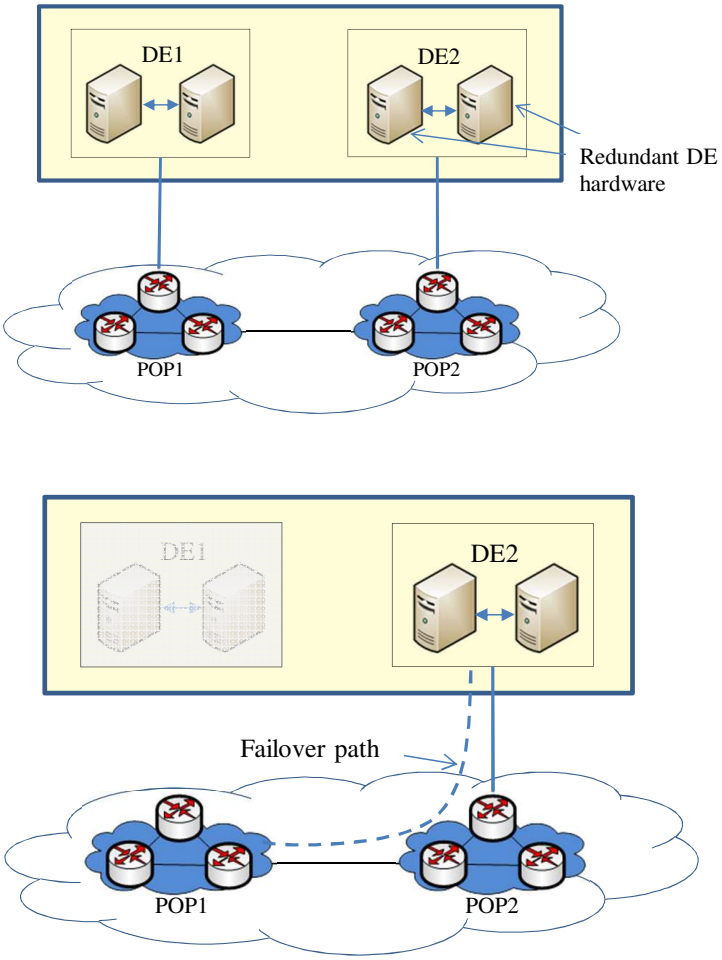


Figure 3.3: Fault-tolerance in LCDP design. Top: (a) Normal operation of the LCDP-managed network with geographic diversity in the decision plane and hardware redundancy at each decision element, Bottom: (b) Mutual failover in LCDP triggered by the failure of DE1

Figure 3.4 shows a high level view of the LCDP architecture where the AS network is logically partitioned into two areas, each controlled by a DE. The partitioning is logical because the two DEs, grouped together, form the logical control plane; exchanging information with each other needed to maintain the network-wide control and maintaining consistent decision-making from a router/switch's perspective. A direct implication of this partition-

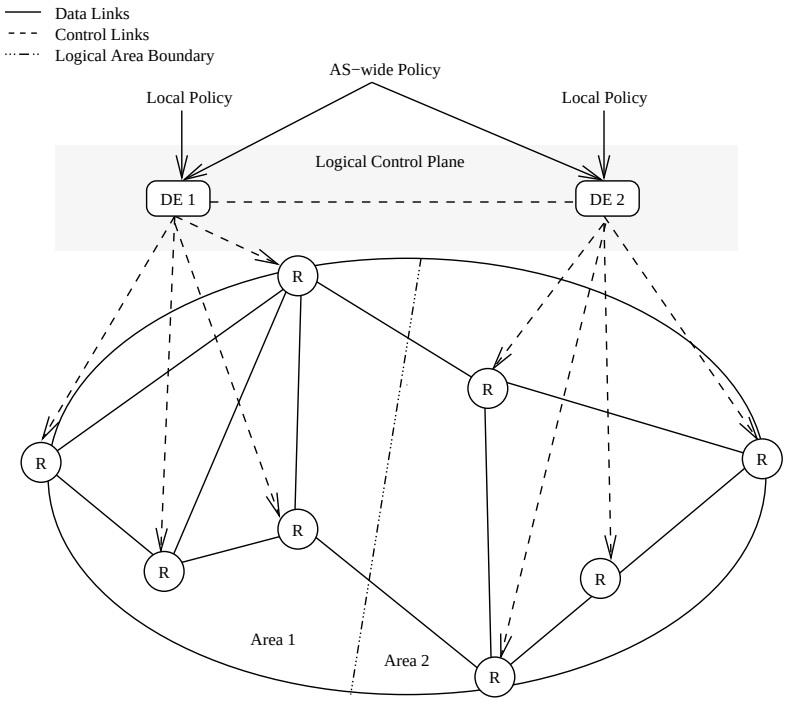


Figure 3.4: High-level overview of logically centralized decision plane with local and AS-wide policies

ing is that a DE will have access to full reachability information about its own area, but may have access to only partial and relevant information about other areas. From a DE's perspective, this means an exchange of the centralized global AS-wide network view with a constrained view comprised of full local-area view and a partial view of the peer-area(s). The extent of the peer-area view depends on the control plane task for which it is needed. For example, link status and reachability information provided by the peer areas through a Link State Advertisement (LSA) packet is sufficient for shortest-path routing. On the other hand, the same peer-area view may not suffice for computing optimal reachability when traffic-dependent link weights [84] are used.

Different potential decision plane functionalities such as traffic load balancing, security threat monitoring, network performance management, may require different levels of peer-area views necessary for their operation. Therefore, we note that while the minimization of

the inter-DE information exchanges is desirable to achieve better scalability of the decision plane, the minimum level of peer-area view can not be determined *a priori* for all tasks that may involve LCDP. Instead of hardwiring the maximum AS-wide view in the design, we allow more variation by believing that the nature of tasks that are added to the decision plane will determine the right balance of peer-area view, and leave the actual split of the AS-wide view to the network designer who is in a better position to determine the necessary peer-area view needed for optimal completion of the decision plane task. This modularity of design to accommodate different design preferences is in line with the principle of modularization along tussle boundaries [85] as our design leaves the actual split of the AS-wide view to the network designer who is in a better position to determine the necessary peer-area view needed for optimal completion of the control plane task.

LCDP design also allows the addition of local-area policies as input to the decision plane. As illustrated in Figure 3.4, AS-wide policy is consistent across all the DEs and may include policies related to security, traffic management, and inter-domain routing. This specification of AS-wide policy is one of the design goals of the 4D network, which specifies that network wide policy should be available to the decision/control layer for optimal decision making. However, the network management may also need control over policy issues related to individual area that does not affect the whole AS network. As an example, a planned maintenance event inside an area may not have network-wide implications if inter-area routing does not change during maintenance. Such local events may be easily controlled with the help of local-area policy giving some control to local network administrators in policy issues that do not require network-wide coordination. Our proposed architecture is in line with the common observation that most large AS networks are partitioned along divisional and geographical boundaries, with each partition operating with some level of independent control. Therefore, the division of network policy into AS-wide and local-area should help in maintaining the natural network organizational structure and result in easier transition to the LCDP architecture.

3.2.4 Summary of LCDP Design

The high-level design of an LCDP based network is shown in Figure 3.5 for an ISP topology spanning the continental US with several POPs. The figure illustrates a logically centralized Decision plane, comprising of three DEs, that is controlling a large ISP network. The ISP network is modeled as having three PoPs, each of which can contain different types of routers, as is commonly seen in ISP networks with backbone, edge, DSL, and other types of routers. In the course of normal operation, as depicted, each DE is seen as controlling a different PoP. The figure also illustrates the few basic assumptions taken in our network model.

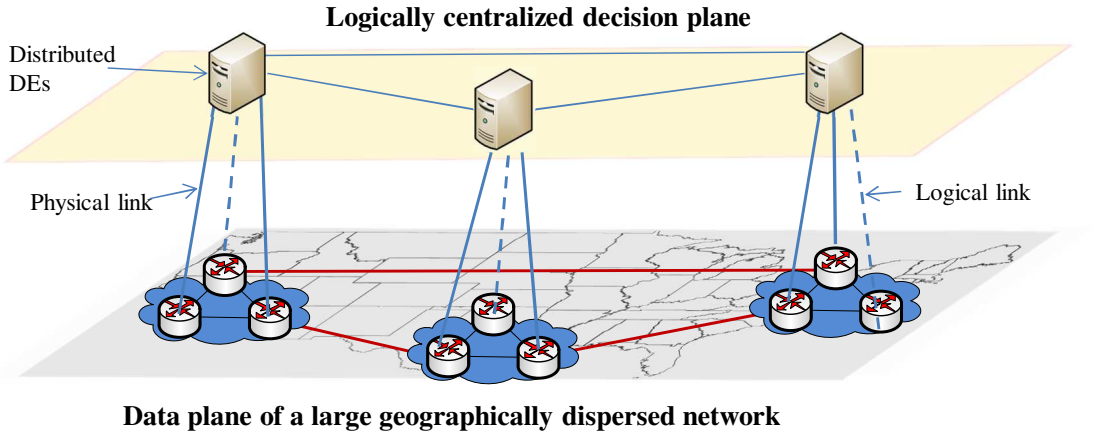


Figure 3.5: Overview of the logically centralized decision plane design

1. The entire network topology is under a single administrative control.
2. The Decision plane is fully connected, i.e. there is a path from each DE to all other DEs that is not dependent on the operation of Data plane.
3. Positioning of DEs corresponds to the natural geographical clustering of routers in the Data plane, e.g. within an ISP POP.

We believe these assumptions are easy to meet in any reasonably large network where control and management is presently an issue. The first assumption is necessary for consistent network-wide management and deserves no further explanation. The use of dedicated out-of-band control paths in the second assumption is in contrast with the in-band paths used in

current IP networks, where data and routing information packets share the same channels. Although it is possible to use the same scheme in logically centralized Decision plane design, we have purposely avoided the potential complexity and network fragility introduced by piggybacking control information over data paths. Our use of out-of-band paths is analogous to the SS7 signaling used in PSTN networks [42] and can be similarly implemented. Use of separate time-slots or wavelength channels for control messages is one way this separation could be accomplished. Finally, our third assumption positions DEs in accordance with the clustering of routers in the underlying data plane [71]. This ensures that latency of Decision plane response, and convergence delay in case of failures, is kept close to minimum.

In LCDP design, each DE is only responsible for computing routing tables for the routers under its direct control, i.e. a subset of the total number of routers in a network. We refer to this (sub)set of routers as an *area* and it marks the extent of a DE's direct control over the network. Moreover, DEs exchange reachability information about their areas and utilize this information in establishing routing paths between different areas. In the case of shortest-paths routing, which we employ for route computation, a path between routers in two different areas must travel the inter-area links between them. This results in optimal routes only under the condition that a similar routing process on the complete topology would have selected the same path. Similar argument also applies to the intra-area routes. It is easy to see that this condition is fulfilled in topologies where distances between routers inside geographical clusters are less than the distance between the clusters. We believe network size and geographical distances between sub-entities in enterprise and ISP networks naturally allow the fulfillment of this condition.

The logically centralized structure of the Decision plane strikes a balance between the extremes of distributed operation of individual routers, as seen in the current data networks, and total centralization, with its inherent scalability and robustness issues. More subtly, it also has the potential to allow easier deployment and transition from a distributed model of operation; as instead of a "forklift" change of the entire networking infrastructure, only a subset of the AS network could be transitioned at a time.

3.3 STRATEGY FOR DECISION ELEMENT PLACEMENT OPTIMIZATION

In this section we examine different strategies for designing the logically centralized decision plane in a network. This includes guidance on the number and positioning of DEs inside an AS network. The optimal positioning of DEs is important as the performance of an LCDP-managed network is heavily influenced by the positioning of DEs in the network. This is especially true for large-scale AS topologies, where large number of network devices and large propagation delays place significant burden on the responsiveness of a centralized decision plane. In such cases, a sub-optimal DE placement can result in unacceptably large route convergence times, as discussed in the next section.

3.3.1 Optimal DE Placement

We can consider several objectives in defining the DE placement optimality: minimization of network cost, convergence delay in case of failure, DE response time, and DE-DE delays can each be considered as optimization objectives. However, these objectives taken together can be contradictory; for example network cost is minimized with a centralized DE, as the single central DE is cheaper than multiple local-area DEs and allows us to gain in economy of scale, while the minimization of DE response time suggests a higher number of DEs to minimize the propagation delays between DEs and routers. In our discussion of the placement strategies, we consider minimization of convergence delay as the primary objective. Minimization of convergence delay is important as it reduces the time for routing to stabilize after any topology changes. Since the actual convergence times are dependent on the routing protocol specifics, we will generalize the worst case convergence delays separately for two different routing strategies: first, where the routing decisions in different areas can be taken independently; and second, those routing strategies that require co-operation among DEs to achieve consistent routing decisions.

Figure 3.6 illustrates both cases where routing decisions in an area are independent or negotiation based. In this figure, three routers $r_1 - r_3$ are serviced by two DEs, e_1 and e_2.

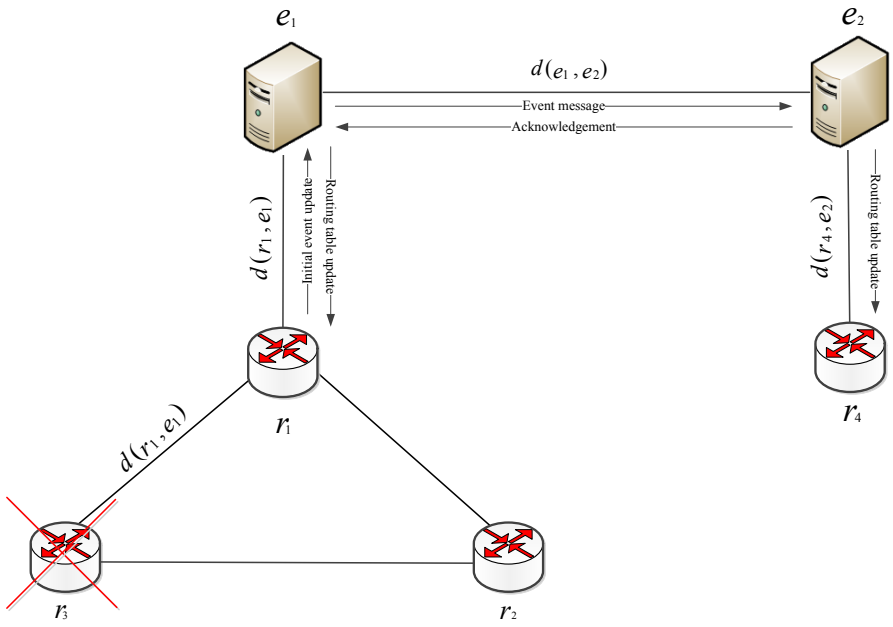


Figure 3.6: Update messages triggered by a router failure in a logically centralized network.

In the case of independent routing decisions, failure of router r_3 triggers a link-state update from router r_1 to e_1, which may involve a timeout at r_1 waiting for keep-alive message on link $r_1 - r_3$. DE e_1 will then compute the updated routing tables, likely after waiting for the expiration of a hold-timer to collect all link-state updates related to the same event, and inform r_1 as well as e_2. e_2 will in-turn compute the updated routing tables for its own area and inform router r_4. In this scenario we assumed a routing policy that takes local decisions at each DE based on available information, without exchanging additional messages than what are needed to disseminate the event information. We note that shortest path routing exhibits this property as each entity takes local decisions without negotiating on the possible choices with other entities. The total time for achieving convergence in this case will be:

$$t_{total} = t_{timeout}(r_1) + d(r_1, e_1) + t_{hold}(e_1) + t_{compute}(e_1) + d(e_1, e_2) + d(e_2, r_2)$$

The values of $t_{timeout}$, t_{hold}, and $t_{compute}$ are protocol-specific and can be assumed to be constant c for a given network size. Therefore, the worst-case convergence delay happens at the maximum of propagation delays:

$$t_{max} = 2d_{max}(r, e) + d_{max}(e, e) + c \tag{3.1}$$

In negotiation based routing, the computation of routing table update requires an exchange of messages between the DEs. Such exchange may be required in routing policies where the objective is to solve a multi-constrained optimization problem and each DE takes part in negotiating the globally optimal solution. In this case, the number of DE-DE exchanges may outnumber the Router-DE messages and so minimizing the convergence delay would require minimizing the aggregate DE-DE delays. We also note that DE-DE exchanges/negotiations may also be necessary in control plane tasks other than routing, and so the applicability of this placement strategy may extend beyond multi-constrained route optimization problems.

In order to minimize the worst-case convergence delay, we need to compute a DE placement strategy that minimizes Router-DE and DE-DE delays. We cast this placement problem as a modification of the capacitated p-median problem [86], where p is the required number of DEs in the AS. The dissemination plane is assumed to be built from shortest paths, as in the case of centralized 4D design. The computation of p is an operational decision that is likely to vary from AS to AS, influenced by several factors:
1. *AS topology* will influence the number of required DEs in several ways. In AS topologies with large geographical distances between routers, the required number of DEs will be higher to constraint the Router-DE delay. Organizational structure and business division boundaries will affect the number of DEs, while higher density of edges in the topology graph may reduce the required number of DEs. 2. *Technological Constraints* include the computational and storage capacities of the DEs. While the computational load for shortest-path routing is within the capacity of a DE build using general-purpose machine [14], the workload on the 4D control plane will increase with the number of control plane tasks. 3. *Robustness Objectives* will require redundancy in the 4D control plane to avoid single points of failure. One such objective is the k-coverage of the routers and other data plane devices, which re-

quires at least k DEs in the AS network. Constraints on the maximum length of Router-DE multi-hop path can also be considered to reduce the susceptibility of Router-DE control path to link failures. 4. *Cost* of the LCDP network would be largely dependent on the number of DEs and so minimizing the cost will minimize the number of DEs. 5. *Performance Objectives* such as the minimization of convergence delay depend on the value of p.

The formulation of the problem requires the value of p as an input. If there is no readily apparent value for p, a network designer can compute the DE positioning for several values of p and compare the outcomes on cost vs. benefit.

3.3.2 Problem Formulation

Let $R = \{r_1, r_2, ..., r_m\}$ be the collection of routers in the AS and p be the number of DEs to be positioned in the network. Let $L = \{l_1, l_2, ..., l_n\}$ be the set of possible locations for the p DEs. We define A as the total number of DE-DE adjacencies, that is $A = p(p-1)/2$. Let d_{ij} be the shortest-path delay between router r_i and a possible DE location l_j. The delay d_{ij} may include queuing and transmission delays, in addition to the propagation delay, especially when multi-hop paths are used between a router and a DE. Let d_{jk} be the shortest path delay between locations l_j and l_k. Let w_i be the measure of r_i's workload, defined as r_i's projected demand on a DE's resources which includes computational, memory, and bandwidth demands. We propose using the size of the routing table as a proxy for the computational demand. Let Q_j be the maximum workload that a DE at location l_j is able to sustain. x_{ij} and y_{jk} are binary variables with $x_{ij} = 1$ if r_i is allocated to the DE at l_j, and $y_{jk} = 1$ if a DE-DE adjacency is identified between l_j and l_k.

In the case of homogeneous DE capacities, the minimum number of DEs needed to cover the network will be:

$$p_{min} = \frac{\sum_{i \in R} w_i}{Q} \tag{3.2}$$

The linear programming formulation given below will indicate the position of DE at site l_j if $y_j = 1$ using the minimization objective in Eqn. (3.1).

$$\text{Min} \sum_{j \in L} \left(\sum_{i \in R} d_{ij} x_{ij} + \sum_{\substack{k \in L \\ k \neq j}} d_{jk} y_{jk} \right) \tag{3.3a}$$

subject to:

$$\sum_{j \in L} x_{ij} = 1 \qquad i \in R \tag{3.3b}$$

$$\sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} y_{jk} = A \tag{3.3c}$$

$$\sum_{\substack{k \in L \\ k < j}} y_{jk} + \sum_{\substack{k \in L \\ k > j}} y_{kj} - (p-1) y_j \leq 0 \qquad j \in L \tag{3.3d}$$

$$\sum_{j \in L} y_j = p \tag{3.3e}$$

$$\sum_{i \in R} w_i x_{ij} - Q_j y_j \leq 0 \qquad j \in L \tag{3.3f}$$

$$x_{ij}, y_{jk}, y_{kj} \in \{0, 1\} \qquad i \in R \;\; j, k \in L \tag{3.3g}$$

Constraint (3.3b) ensures that a router is assigned to exactly one DE. Constraint (3.3c) is used to guarantee the correct number of DE-DE adjacencies in the objective function. Constraint (3.3d) forbids adjacencies for locations where a DE is not present. Constraint (3.3e) limits the total number of DEs to p. Finally, by Constraint (3.3f) we ensure that the total assigned workload at a location does not exceed the available capacity at that location. We observe that this formulation's objective is more sensitive to the aggregate delays between routers and DEs, in comparison to the DE-DE delays, as the number of routers is greater than the DEs. Therefore, there will be more terms where $d_{ij}x_{ij}$ is positive as compared to terms where $d_{jk}y_{jk}$ is positive. This will increase the sensitivity of this formulation to router-DE delays. For the routing strategies where the routing decision at a DE may not be taken independently, we minimize the DE-DE delay while bounding the maximum router-DE delay by a constant B. The new LP formulation, with the router-DE delay bounded by

B in Constraint (3.4g), is:

$$\text{Min} \sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} d_{jk} y_{jk} \tag{3.4a}$$

subject to:

$$\sum_{j \in L} x_{ij} = 1 \qquad i \in R \tag{3.4b}$$

$$\sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} y_{jk} = A \tag{3.4c}$$

$$\sum_{\substack{k \in L \\ k < j}} y_{jk} + \sum_{\substack{k \in L \\ k > j}} y_{kj} - (p-1) y_j \leq 0 \qquad j \in L \tag{3.4d}$$

$$\sum_{j \in L} y_j = p \tag{3.4e}$$

$$\sum_{i \in R} w_i x_{ij} - Q_j y_j \leq 0 \qquad j \in L \tag{3.4f}$$

$$d_{ij} x_{ij} \leq B \qquad i \in R \;\; j \in L \tag{3.4g}$$

$$x_{ij}, y_{jk}, y_{kj} \in \{0, 1\} \qquad i \in R \;\; j, k \in L \tag{3.4h}$$

In addition to the constraints considered in the earlier formulations, we can consider another constraint to balance the computational load between the DEs. This will ensure that the workload is balanced among the DEs at the beginning of network operation. However, as the network topology changes as a result of usual network dynamics, another mechanism will be needed to maintain a balanced assignment of data plane devices to the decision plane. An adaptive router assignment algorithm is discussed in Section 4.3 to address this issue. Here, we discuss a third problem formulation that introduces a load balancing constraint to balance the DE work-loads using the average load, L_{avg}, and a load balancing parameter $\Delta \geq 1$.

$$L_{avg} = m / \sum_{e_j} Q_j \qquad 0 < L_{avg} \leq 1$$

The new optimization problem can be formulated as following:

$$\text{Min} \sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} d_{jk} y_{jk} \tag{3.5a}$$

subject to:

$$\sum_{j \in L} x_{ij} = 1 \qquad i \in R \tag{3.5b}$$

$$\sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} y_{jk} = A \tag{3.5c}$$

$$\sum_{\substack{k \in L \\ k < j}} y_{jk} + \sum_{\substack{k \in L \\ k > j}} y_{kj} - (p-1) y_j \leq 0 \qquad j \in L \tag{3.5d}$$

$$\sum_{j \in L} y_j = p \tag{3.5e}$$

$$\sum_{i \in R} w_i x_{ij} - Q_j y_j \leq 0 \qquad j \in L \tag{3.5f}$$

$$d_{ij} x_{ij} \leq B \qquad i \in R \;\; j \in L \tag{3.5g}$$

$$\sum_{i \in R} w_i x_{ij} \leq \lceil \Delta L_{avg} Q_j \rceil \qquad \forall e_j \in E \tag{3.5h}$$

$$x_{ij}, y_{jk}, y_{kj} \in \{0, 1\} \qquad i \in R \;\; j, k \in L \tag{3.5i}$$

Load balancing trade-offs and the constraint 3.5h are discussed in Section 4.3.1.

All of the given formulations are Integer Linear Programs (ILP) and are similar to the type of problems referred to as "capacitated p-median" in operation research and facility optimization literature. The distinguishing characteristic of this type of problem is that it seeks to optimize the location of a set p of facilities— DEs in our context — against the constraints of distances, service loads, and location capacities. Capacitated p-median problems are known as NP-hard [87, 88]. However, these can be solved using standard ILP solution techniques such as Branch-and-Bound methods [89], Branch-and-Price methods [90], and Cutting Plane techniques [91]. Other approaches for solving p-median problems have also been developed and include column generation [88], simulated annealing [92], tabu search [87], and genetic algorithms [93].

Since the 4D architecture proposes the separation of data and dissemination paths, the bound on maximum delay (B) and DE work-load (K_j) provided in the two formulations will not be affected by the dynamic routing choices in normal conditions. However, failures of dissemination paths, due to failures in either control or data planes, can lead to discovery of new dissemination paths that violate the bounds on B and K_j. The magnitude of deviation from these bounds will depend on the connectivity of the AS topology graph.

3.3.3 Evaluation

In this section, we provide computational results for the optimal DE placement problem, 3.3, given in the previous section. We investigated five different AS topologies from the Rocketfuel project [82] and utilized the GNU Linear Programming Kit (GLPK) [94] to solve the integer linear programs for DE placement using Branch-and-Bound procedure [89].

Rocketfuel reports latencies and inferred weights between pair of vertices (routers), and we used the latency values between vertices i and j as the measure of shortest-path delay d_{ij} in our model. Since Rocketfuel measurements were made online, this measure of delay contains average queuing delay between the pair of vertices in addition to the propagation delay. The AS topologies were checked for connectivity and the largest connected component was utilized when full connectivity was not found in the instance graph. The number of routers m, the maximum shortest-path delay d_{max}, and the average shortest-path delay d_{avg} for the instances are shown in Table 3.1.

To limit the size of the problem, we considered $n = 10, 15, 20$ most central vertices as possible locations L for the DEs. We used the "betweenness" of a vertex as the measure of centrality and goodness of choice when a DE is located at that vertex. Betweenness is a measure of centrality, commonly used in social networks and network survivability analysis, that values those vertices more which occur on shortest paths (geodesic) between many other vertices. Therefore, a vertex with higher betweenness provides a better choice for the placement of a DE, when the number of DEs need to be minimized. Betweenness is formally defined for a vertex v as [95]:

Table 3.1: Rocketfuel topology summary

AS Number	Vertices	d_{max}(ms)	d_{avg}(ms)
1221	104	54	7.82
1755	87	47	6.25
3257	161	83	7.77
3967	79	105	11.93
6461	138	137	17.43

$$C_{B(v)} = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

σ_{st} is the number of shortest paths from s to t and $\sigma_{st}(v)$ is the number of shortest paths that v lies on.

Figures 3.7, 3.8, 3.9 show the plots of the average delay between a router and a DE for different Rocketfuel topologies as a function of p for $n = 10, 15, 20$ possible DE locations. The plots indicate that investigation of only a small subset of most central locations is sufficient in locating optimal DE placement. It can be seen from the figures that the "knee" of the average delay contours occurs around $p = 3$ to 4 in the tested topologies. Reduction in the average delay is evident in comparison to the observed delays in Table 3.1. This shows that a distributed decision plane with even a few DEs will give much better route convergence delays as compared to a centralized design.

3.4 SUMMARY

This chapter presented the motivation and design vision for a logically centralized decision plane that is physically distributed over a set of controllers, or DEs, in a network. We argued

that, in contrast with the current network control approaches that are either fully centralized or fully distributed, logical centralization provides network operators an opportunity to customize the level of centralization that makes sense in the context of the design and requirements of their networks. We expect that this approach of providing network operators with a customizable design of network decision plane will be very useful in the management of large enterprise networks and will provide the management scale needed for the Internet to meet the future growth challenges.

The number and physical placement of the decision elements is a key design consideration in logically centralized networks. This chapter also investigated the physical design of LCDP networks, and provided techniques that can be used to optimize the physical placement of DEs. The placement of DEs was optimized by using a modified p-median formulation approach that aimed at maximizing decision plane responsiveness. We used real ISP topologies from Rocketfuel project to evaluate our approach and found that even in very large network topologies, a small number of decision elements are sufficient in ensuring decision plane responsiveness that is similar to the performance of distributed route computation.

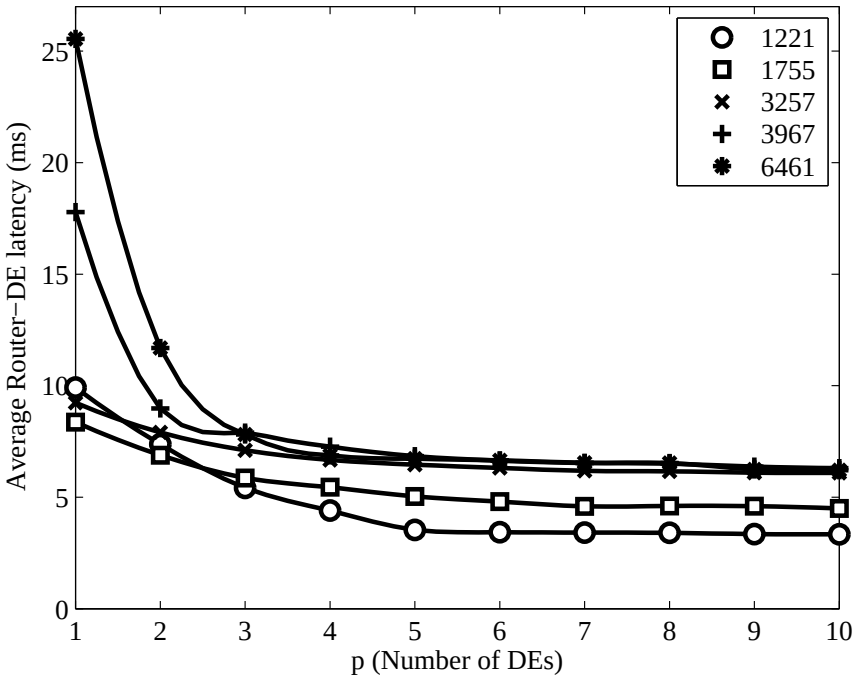


Figure 3.7: Plot of average Router-DE delay for $n = 10$

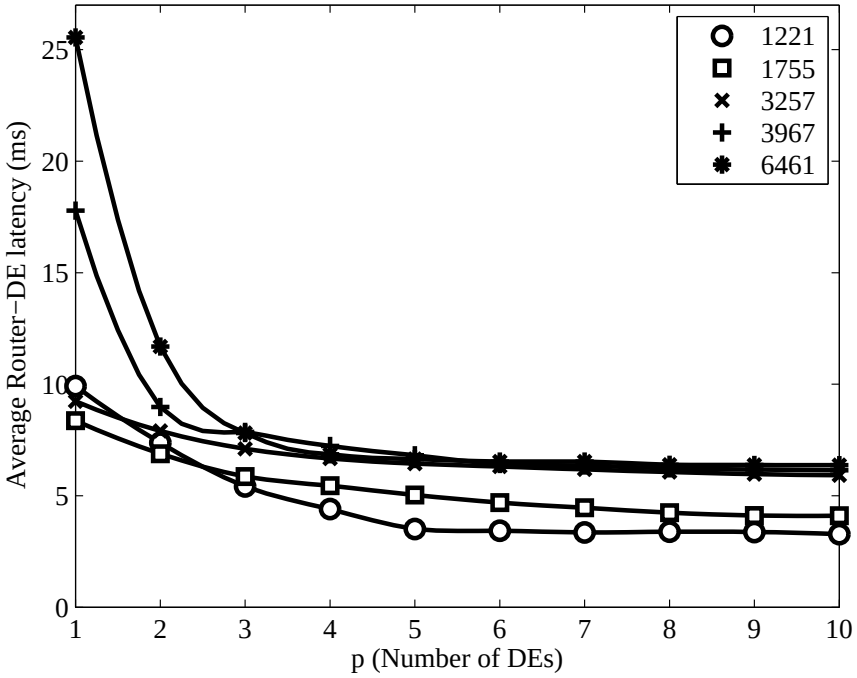


Figure 3.8: Plot of average Router-DE delay for $n = 15$

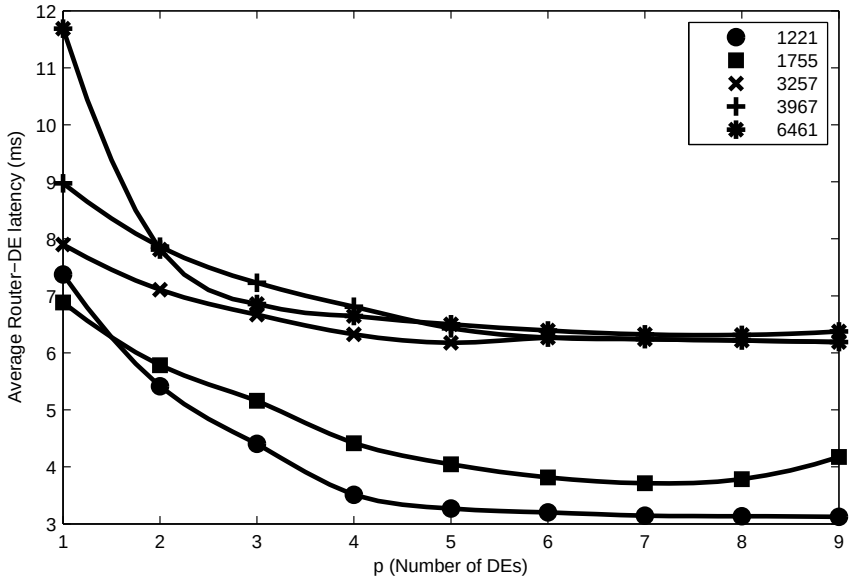


Figure 3.9: Plot of average Router-DE delay for $n = 20$

4.0 MECHANISMS FOR SCALABLE AND ROBUST DECISION PLANE OPERATION

This chapter will present a design for logically centralized decision plane with emphasis on design scalability and robustness to failures. Scalability is a key factor that needs to be explicitly considered in any large-scale network design, especially so within the context of large ISP and enterprise networks primarily considered in this work. Robustness to failures has been one of the original design requirements of Internet and this characteristic needs to be preserved in any future design. In the following sections, we build upon the network model presented in Chapter 3 and present an algorithm and a protocol that together allow fast convergence to failures of a logically centralized decision plane.

4.1 OVERVIEW

The design of an efficient and robust decision plane requires careful consideration of the design efficiency, scalability, and robustness. A physically centralized decision plane design was investigated in [13, 14], where the entire autonomous system was controlled by a single Decision Element (DE) and replication of DEs was used to ensure decision plane robustness to DE failures. An alternative design approach was identified in [71], where the logically centralized decision plane was distributed over physically independent DEs. In this design, each DE controls a subset of the entire network and works collaboratively with other DEs to achieve overall network control. This approach of logically centralized decision plane design tries to balance the extreme design positions of total distribution of network control, as seen in the case of current IP networks, and total physical centralization. However, it is

also important to ensure that the reliability of the physically distributed control approach matches or exceeds the reliability offered by today's distributed architecture.

In this chapter, we focus on the design of logically centralized clean-slate decision plane as the basis for developing an efficient, robust, and reliable network control architecture. We argue that the decision plane design should be based on meeting the following objectives:

- **Scalability:** The decision plane must be scalable to network size in terms of the number of routers.
- **Robustness:** The design should be dynamically adaptable to failures at both decision and data planes.
- **Optimal convergence:** Total response time of the decision plane to any network event must be minimized, and the protocol operating at the decision plane should be able to converge quickly enough to operate on the time-scale of events happening at the data plane, e.g. router/link failures.

Achieving these objectives requires the development of a decision plane protocol (DPP) that maintains a network-wide state across the set of physically distributed DEs, and presents a uniform interface to the network switches or routers[1]

Furthermore, the DEs and their assigned routers must respond swiftly to events such as failures and traffic surges. This requires that the delay between the DEs and their assigned routers be minimized. This chapter addresses these design requirements and presents a decision plane where a set of DEs, each governing a subset of routers, collaboratively maintains a network-wide state to support network-wide routing decisions.

The main contribution of in this chapter is the design of a scalable *logically centralized and physically distributed* decision plane. The first building block in our design is the formulation of an optimization problem focused on efficient assignment of routers to DEs. The solution of this problem leads to an algorithm that minimizes network delay between the DEs and their assigned routers while balancing the load at the DEs. This algorithm is then used in the proposed protocol that is responsible for the operation of logically centralized decision plane.

[1]We use "router" as a generic label for routers or switches, while "DE" is used to represent Decision Elements.

4.2 TRADE-OFFS IN DECISION PLANE DESIGN

Robustness of the decision plane is dependent on the mechanisms employed to ensure its continued functioning in case of failures. While the decision plane routing logic deals with failures happening at the data plane, the mitigation of failures at the decision plane is dependent on its own design. An approach to this problem was presented in [14], where the decision plane was designed to be physically centralized and multiple hot-standby DEs were used to increase its robustness in case the current "master" DE fails.

In contrast, a DE in a logically centralized decision plane is not required to control the entire AS; only a subset of the total number of routers are under the control of a single DE. Any DE failure would therefore orphan the routers under its control. This calls for a scheme that reassigns orphaned routers to the functional DEs so that network control is reinstated.

This assignment of routers takes place both at network bootstrap and as a result of DE failures. It involves a trade-off in minimizing routing convergence delay, response time, and load balancing at the decision layer. The routing convergence delay — transient time period between DE failure and orphaned routers' reception of new routing assignments — represents loss of management control over the orphaned devices, and must be minimized. Similarly, in normal operation the response time of decision plane also needs to be minimized. In both cases, aggregate router-DE delay provides a desirable metric for the minimization objective. Additionally, large variation in DE work-loads can result in slower decision plane response in parts of the network and increased potential for DE failures, suggesting a need for load balancing at the decision plane. Therefore, the optimality of router assignment will be based on minimizing the aggregate router-DE delay while limiting the variance in DE work-loads.

Assignment mechanism is also constrained in a unique way as any router assignment must adhere to the underlying physical data plane topology. Specifically, since a DE only controls the routers in its own logical area, the assignment mechanism must avoid any assignment that involves the usage of inter-logical area paths between routers belonging to the same logical area. This condition is necessary to ensure that routers in a logical area can be governed locally without requiring AS-wide network knowledge. Therefore, there must be a physical path between routers that are assigned to the same DE that does not involve

any links or routers not totally contained within the same logical area. We refer to this condition as the contiguity constraint and Figure 4.1 illustrates a simple example where the assignment that is optimal in terms of delay and load balancing objectives does not satisfy the contiguity requirement.

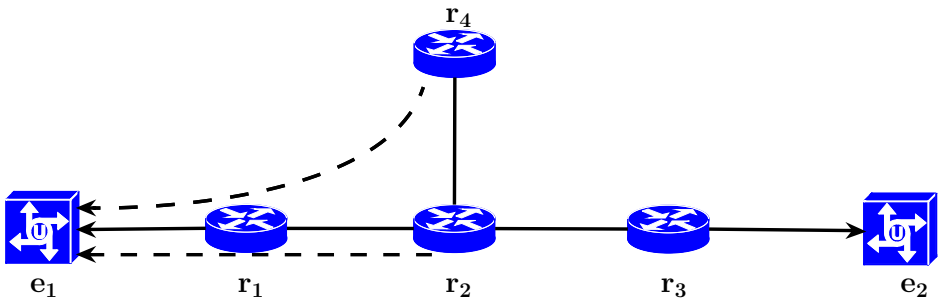


Figure 4.1: Effect of contiguity constraint on a sample topology where multi-hop router assignments are indicated by dashed lines. The (infeasible) assignment of router r_4 to DE e_2 would have resulted in minimal delay and optimal load-balancing.

Trade-offs also exist between complexity of a recovery scheme and the desired level of robustness. For example, we can generalize a simple scheme of using backups as proposed in [14, 56], where each router is statically configured with a primary and an ordered list of standby DEs. Failure of the primary DE automatically results in the assignment of its orphaned routers to their highest-ranked functional DEs. However, it is easy to show that this scheme can lead to uneven DE work-loads in case of multiple DE failures, potentially causing severe performance degradation. Moreover, the underlying network topology, on which any static assignment is based, may change due to the dynamics of network operation, such as link and device failures — potentially making a static order of assignment infeasible or highly inefficient. Consequently, a static assignment will have to be updated to ensure its applicability and validity with the dynamically changing network topology. These shortcomings of static ordering schemes suggest that it is desirable to have an adaptive mechanism, that can assign routers to feasible DEs while, 1., balancing the DE workload and, 2., minimizing the physical constraint on decision plane response time, i.e. the propagation delays between routers and DEs. In the following section we describe our design of such adaptive router assignment mechanism.

4.3 ADAPTIVE ASSIGNMENT OF DATA PLANE DEVICES

Let $R = \{r_1, r_2, ..., r_m\}$ be the collection of routers in a AS, assumed to be homogeneous in terms of their demands of decision plane resources, and $E = \{e_1, e_2, ..., e_n\}$ be the set of n functional DEs in the network. For any r_i, $N(r_i)$ denotes the set of routers in physical open neighborhood of r_i, i.e. r_i and all of its physically adjacent routers. We define $A(e_j)$ to be the set of routers assigned to e_j and A as the adjacency matrix of router assignments for all DEs in E, which is the output of the assignment problem. Let $x(r_i, e_j)$ be a binary indicator variable defined as $x(r_i, e_j) = 1 \iff e_j \leftarrow r_i$. Let $d(r_i, e_j)$ be the minimum delay between router r_i and a DE e_j, and $D[d(r_i, e_j)]_{m\text{xn}}$ be the matrix of all such delays. Let $L_j = \sum_{r_i \in R} x(r_i, e_j)$ be the load on DE e_j and Q_j be the capacity, i.e. the maximum number of routers, that e_j is able to govern.

We assume that information about the network topology, specifically router adjacencies and delay, would be available to the decision plane as part of the service offered by the discovery and dissemination planes of 4D architecture. Use of source routes [14, 56] is one method by which such information can be collected, and Section 4.4.2 discusses the protocol primitives that can be used for inter-layer communication. However, the design specifics of discovery and dissemination planes are beyond the scope of this work.

4.3.1 ILP Formulation

From the discussion of the previous section, the objective of the assignment problem is to assign routers in R to DEs in E in such a way that aggregate delay between routers and their assigned DEs is minimized, while ensuring that the DE workload is balanced. Formally, we define our objective function as

$$\sum_{e_j \in E} \sum_{r_i \in R} d(r_i, e_j) x(r_i, e_j)$$

We introduce a constraint to balance the DE work-loads using the average load, L_{avg}, and a load balancing parameter $\Delta \geq 1$.

$$L_{avg} = m / \sum_{e_j} Q_j \qquad 0 < L_{avg} \leq 1$$

The optimization problem can be formulated as the following ILP:

$$\text{Minimize} \sum_{e_j \in E} \sum_{r_i \in R} d(r_i, e_j) x(r_i, e_j) \tag{4.1}$$

s.t.

$$\sum_{e_j \in E} x(r_i, e_j) = 1 \qquad \forall r_i \in R \tag{4.2}$$

$$\sum_{r_i \in R} x(r_i, e_j) - Q_j \leq 0 \qquad \forall e_j \in E \tag{4.3}$$

$$L_j \leq \lceil \Delta L_{avg} Q_j \rceil \qquad \forall e_j \in E \tag{4.4}$$

$$\sum_{r_k \in N(r_i)} x(r_k, e_j) \geq x(r_i, e_j) \qquad |A(e_j)| \geq 1, \forall r_i \in R \tag{4.5}$$

$$x(r_i, e_j) \in \{0, 1\} \qquad \forall r_i \in R, e_j \in E \tag{4.6}$$

The objective function minimizes aggregate delay between routers and their assigned DEs. Constraint (4.2) ensures that each router in R is assigned, constraint (4.3) ensures that the DE workload capacities are not violated, and constraint (4.5) imposes the contiguity requirement.

The load balancing constraint (4.4) is weighted by a parameter, Δ, which controls the maximum deviation of a DE's normalized workload from the average normalized workload for all DEs. Setting $\Delta = 1$ would force workloads of all DEs to be exactly equal to the average normalized workload, or in other words each DE will have the same fractional utilization of its capacity as all others. In case of homogeneous DE capacities this translates to an equal workload for all DEs. On the other hand, $\Delta > 1$ allows the normalized workload of at least one DE to be higher than the average by $(\Delta - 1) * 100$ percentage.

The value of Δ also dictates the trade-off between the objectives of minimum aggregate delay and load balancing as it changes the feasible set of solutions. A large value of Δ optimizes a solution for the objective of minimizing aggregate delay, while a tighter constraint will show significant trade-off in favor of load balancing. The addition of a hard constraint for load balancing comes at the cost of reduced feasibility where optimal solutions could be infeasible because of a choice of Δ which is too low. This situation is likely to arise in tightly constrained problems especially in the event of reduced capacity as a result of DE

failures. However, the dependence of constraint (4.4) on the average normalized workload ensures that the formulation dynamically adapts to failures, as a DE failure lowers the total available capacity thereby increasing right hand side of the constraint. This will result in higher workload shares for the remaining functional DEs to accommodate the orphaned routers. If the total capacity of the remaining DEs is less than the workload offered by the data plane, no feasible solution will exist for the problem.

Our approach is different from the traditional load balancing method of minimizing the maximum load, and provides better control to a network operator while ensuring robust and efficient operation of the decision plane. The sub-problem with only the minimum delay objective and constraints (4.2), (4.3) and (4.6) is commonly referred to as Terminal Assignment Problem [96], which is NP-complete in case of non-homogeneous router weights and DE capacities [97].

4.3.2 Two-phase Router Assignment Algorithm

We construct a two-phase exact algorithm to solve the optimization problem. The first phase of the algorithm constructs an ordering of routers, S, where S is the sorted order of minimum delay assignments for each router, and greedily assigns routers in the order of S to their closest (min-delay) feasible DEs, if such assignments are possible. To meet the contiguity constraint (4.5), a router r_i's assignment is made to the closest DE e_j if $d(r_i, e_j)$ is strictly less than the delay between r_i and any other DE and e_j has slack capacity. On the other hand, if there are other DEs at same delay from r_i as e_j, r_i is assigned to a feasible DE that has an existing assignment in $N(r_i)$. Otherwise, r_i is kept unassigned.

The goal of the first phase of algorithm is to make all feasible lowest-cost assignments that can be made without changing any previously made assignments. This phase constructs an optimal solution for the assigned routers. Any routers that remain unassigned after the first phase are assigned by the second phase using a branch exchange algorithm that iteratively accommodates previously unassigned routers, while maintaining feasibility of the solution. Our solution is $O(m^2n)$ in the worst case, and finds optimal solution to the assignment problem if it exists.

4.3.2.1 Greedy Phase We utilize a greedy heuristic to assign routers to DEs while maintaining the feasibility of solution. Since, by definition, a greedy approach does not make any changes to its local decisions, the order in which decisions are taken becomes important. Our approach considers routers in the order of lowest assignment costs for each router. Assignments are made only with a feasible min-delay DE, where feasibility is determined by the constraints given in Section 4.3.1. Figure 4.2 describes the definitions and operation of this phase.

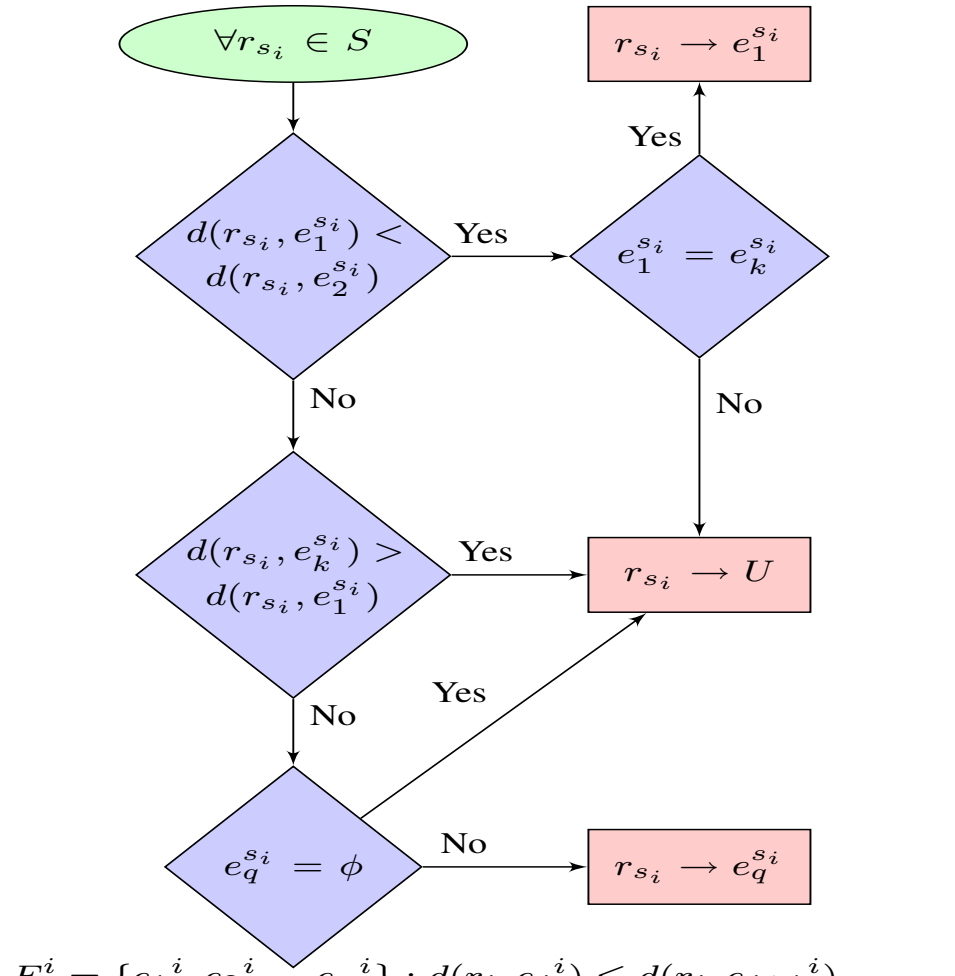
Lemma 1. *Let $x(r_{s_i}, e_k^{s_i})$ be an assignment made in the greedy phase. By construction, $d(r_{s_i}, e_k^{s_i}) \leq d(r_{s_i}, e_j^{s_i})\ \forall e_j^{s_i} \in E^{s_i}$ i.e. $e_k^{s_i}$ must be the minimum cost feasible assignment for r_{s_i}.*

The algorithm explicitly checks a potential assignment against the capacity (4.3) and load balancing (4.4) constraints, while implicitly meeting the contiguity constraint (4.5) according to the following Lemma:

Lemma 2 (Greedy Phase meets constraint (4.5))**.** *Since router assignments are done strictly in the order of min-delay, it suffices to show that routers assigned in this order will meet the contiguity constraint. We prove this Lemma by induction on the assignment of a router r_{s_i}: If $A(e_1^{s_i}) = \phi$, the Lemma trivially holds as r_{s_i} must be directly connected with $e_1^{s_i}$ by Lemma 1. For the case of $A(e_1^{s_i}) \neq \phi$, we assume that Lemma holds for $i-1$ assignments and r_{s_i} is the i^{th} assignment that violates the Lemma, implying $\exists r_a \notin A(e_k^{s_i})\ \forall r_a \in N(r_{s_i})$*

Conditioning on r_a, we observe that there must be a path from r_{s_i} to $e_k^{s_i}$ which passes through r_a. Hence, $d(r_{s_i}, e_k^{s_i}) = d(r_{s_i}, r_a) + d(r_a, e_k^{s_i})$ which implies $d(r_a, e_k^{s_i}) < d(r_{s_i}, e_k^{s_i})$. Therefore, r_a must have been picked by the algorithm before r_{s_i} and since $e_k^{s_i}$ is a feasible choice for r_{s_i} it must have been a feasible choice for r_a. This implies r_a is assigned to an arbitrary DE e_1^a where $e_1^a \neq e_k^{s_i}$ and $d(r_a, e_1^a) < d(r_a, e_k^{s_i})$. By substitution, it can be seen that this results in $d(r_{s_i}, e_1^a) < d(r_{s_i}, e_k^{s_i})$, thus violating Lemma 1. Therefore, i^{th} assignment must be valid.

4.3.2.2 Exchange Phase The greedy phase makes all the feasible min-cost router assignments that can be made without changing any existing assignment. Consequently, as-



$E^i = \{{e_1}^i, {e_2}^i, .., {e_n}^i\} : d(r_i, {e_j}^i) \leq d(r_i, {e_{j+1}}^i)$

$S = \{r_{s_1}, r_{s_2}, .., r_{s_m}\} : d(r_{s_i}, {e_1}^{s_i}) \leq d(r_{s_{i+1}}, {e_1}^{s_{i+1}})$

$U = \{\text{Set of unassigned routers}\}$

$k = \text{Index of the first feasible DE in } E^i$

$e_q^{s_i} \in E^{s_i} \quad k \leq q < n :$

$\exists r_a \in N(r_{s_i}), A(e_q^{s_i})$

$d(r_{s_i}, e_q^{s_i}) = d(r_{s_i}, e_1^{s_i})$

Figure 4.2: Greedy phase of the router assignment algorithm

signment of an unassigned router after the greedy phase's completion may involve a trade-off between sub-optimal assignment to available DEs or reassignment/exchange of already assigned routers to allow a lower cost assignment. Therefore, in order to ensure optimality

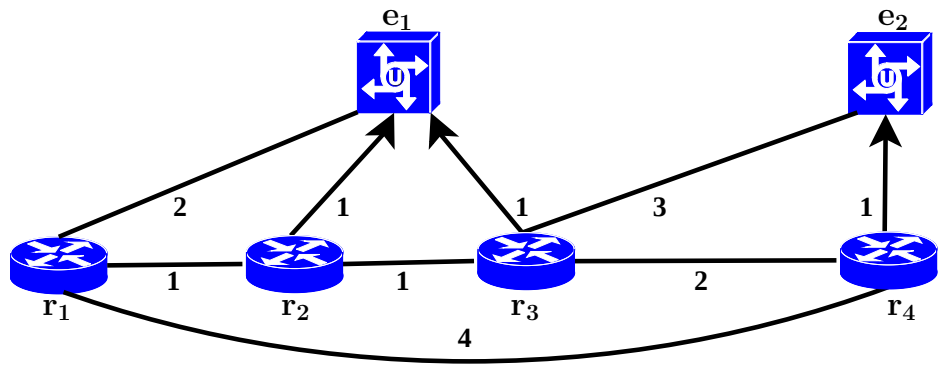
of the solution, the assignment mechanism must be able to find the lowest-cost set of exchanges that allows the assignment of an unassigned router. This mechanism is provided by the exchange phase, which utilizes a branch-exchange algorithm, similar in design to the method described in [96], to construct an auxiliary graph of the network and uses shortest path algorithm for computing lowest-cost assignments.

In simple terms, auxiliary graph represents the feasible combinations of router assignment exchanges between DEs, weighed by the cost of such exchanges. The min-cost path through the graph represents the min-delay assignment for a previously unassigned router. Therefore, edges of the graph represent possible feasible exchanges (and new assignments) between DEs which, themselves, are represented by the graph's vertices. Similar to the greedy phase, feasibility of any exchange or new assignment depends on conformance to the constraints presented in Section 4.3.1. Auxiliary graph is constructed according to the following rules:

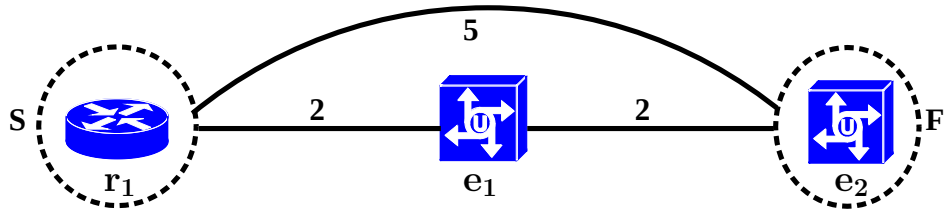
- There are two special vertices S and F that represent the source and destination vertices for the shortest path computation. The shortest path from S to F, at each iteration of exchange phase, provides the lowest cost assignment of one unassigned router.
- There are additional vertices, $Y = Y_1, Y_2, .., Y_k$, each corresponding to a DE without any slack capacity.
- There is an edge (S, Y_k) corresponding to potential assignment of an unassigned router $Y_k \leftarrow r_i : \exists r_a \in A(Y_k),\ r_a \in N(r_i)$ with an edge weight $d(r_i, Y_k)$.
- There is an edge (Y_k, Y_l) corresponding to a router r_i at the border of Y_k and Y_l's logical areas, such that $x(r_i, Y_k) = 1,\ \exists r_a \in A(Y_l), r_a \in N(r_i)$ and the weight $d(r_i, Y_l) - d(r_i, Y_k)$ is positive.
- There is an edge (Y_k, F) corresponding to a router r_i's feasible re-assignment from Y_k to a DE e_j with slack capacity. The weight of this edge is $d(r_i, e_j) - d(r_i, Y_k)$. If there are no DEs with slack capacity, an auxiliary graph can not be created and the algorithm will terminate.
- There is an edge (S, F) with weight $d(r_i, e_j)$ for $e_j \leftarrow r_i$.

Lemma 3 (The auxilary graph has no negative cycles)**.** *There can not be any negative cycles involving S and F vertices, and so it only remains to be shown that the vertices in Y do not*

have any negative cycles between them. We observe that only edges with positive weights are allowed between vertices in Y, and since a negative cycle implies edges with negative weights, the Lemma is proven by construction.



(a) Topology with r_1 unassigned



(b) Auxiliary graph where $(S, e_1) = e_1 \leftarrow r_1$ and $(e_1, F) = e_2 \leftarrow r_3$

Figure 4.3: Operation of the exchange phase on a network example where $\Delta = 1$ and edges are annotated with delay values. The min-cost assignment is along $(S, e_1), (e_1, F)$

Dijkstra's shortest path algorithm is used to compute the shortest path on the directed auxiliary graph from vertex S, which represents an unassigned router, to F, which represents DEs with slack capacity. Lemma 3 establishes that Dijkstra's algorithm, which can only be used in graphs with no negative cycles, is applicable to the auxiliary graph. This shortest path represents the minimum cost set of exchanges that are needed to assign a previously unassigned router. The auxiliary graph is updated after the assignment and the process repeated until all routers have been assigned, or no DE remains with slack capacity.

Figure 4.3 shows the operation of exchange phase for a simple network example. The network topology and assignments after the completion of greedy phase are shown in Fig-

ure 4.3a, where r_1 is unassigned as it can not be assigned to its closest DE, e_1, without violating the strict ($\Delta = 1$) constraint on load balancing. Figure 4.3b shows the construction of the auxiliary graph using the auxiliary graph construction rules on the network topology. In this figure, the shortest-path from the source node, S, to the destination node, F, represents the min-cost assignment for the unassigned router, r_1. The min-cost assignment, $((S, e_1), (e_1, F))$, assigns r_1 to e_1 while reassigning r_3 from e_1 to e_2, at a total cost of 4. The other possible path from S to F represents the direct assignment of r_1 to e_2, at a higher total cost of 5. Since only one r_1 was unassigned in Figure 4.3a, the algorithm will terminate after its assignment.

4.3.3 Analysis

The exchange phase of the algorithm tries to iteratively assign all of the routers that were unassigned at the end of the greedy phase. In networks where $\sum_{e_j} Q_j \geq m$, i.e. when enough capacity exists at the decision plane to handle all the routers in the network, there will always be at least one DE with slack capacity to allow the formation of auxiliary graph. More specifically, vertex F, which serves as the destination vertex for shortest path computation on the auxiliary graph, can only be reachable from the source vertex, S, if a DE with slack capacity exists in the network. In cases where the construction of auxiliary graph fails, i.e. when $\sum_{e_j} Q_j \leq m$, the algorithm will terminate with the set of already computed router assignments as output. This set will contain the maximum number of router assignments that can be made without violating the constraints on DE capacities. Otherwise, the algorithm will terminate after assigning all m routers in the network.

The greedy phase of the algorithm is $O(m)$. The exchange phase's complexity is dependent on the shortest path computation, with worst case complexity of $O(n^2)$. The exchange phase calls Dijkstra's algorithm for each unassigned router, resulting in an overall worst case complexity of $O(mn^2)$. In reality, the greedy phase assigns most of the routers, and the few unassigned routers in tightly-constrained DE failure scenarios each require one iteration of the exchange phase. This results in average-case complexity of $O(m + kn^2)$, where $k \ll m$. Also, since the number of routers in a network are expected to be much higher than the

number of DEs, i.e. $m \gg n$, complexity of the scheme is dominated by the complexity of greedy phase, resulting in very fast run-times e.g. less than $3.5s$ on average in a network with $(m, n, \Delta) = (1500, 10, 1.0)$ as described in Section 4.5.

4.4 DPP PROTOCOL FOR DECISION PLANE OPERATION

In this section we discuss the design of an experimental protocol for the operation of logically centralized decision plane using the router assignment algorithm. A discussion of the main functional requirements of DPP protocol is presented, followed by a description of the protocol structure and states, and finally we discuss how the protocol interacts with other layers of the 4D architecture.

4.4.1 Functional Requirements

The protocol operating at the decision layer is responsible for management of DEs in providing a uniform network-wide decision plane. To effectively meet the design goals specified in Section 1, the design needs to conform to the following basic functional requirements:

- Robustness to multiple failures in the decision and data planes must be insured. This implies a design that incorporates redundant control logic and storage of network state.
- Any pre-configuration of protocol parameters should be minimized and the protocol must be able to operate without constant human intervention.
- Protocol must be easily extensible and evolvable to include additional functionalities.
- To improve scalability of the decision plane, the protocol must distinguish between events which have network-wide significance vs. events which have their impact limited within a local DE's control boundaries. For example, failure of a redundant link totally contained within a logical area may not have AS-wide significance, while failure of a backbone link connecting two different logical areas might require re-computation of routing matrices at multiple DEs to redirect traffic away from the affected link.

- The protocol must be able to deal with synchronization issues expected in the control of a large geographically-dispersed AS.

These requirements are not meant to be exhaustive but to serve as a guideline for the protocol design.

4.4.2 Protocol Design

The functional requirements of the previous section provide a basis for the design of DPP protocol where we incorporate the following salient design features:

Leader Election: Router assignment algorithm is computed only by the DE which has been chosen to act as leader. We utilize a simple leader election protocol based on unique pre-configured DE identifiers. The leader election protocol is used at network bootstrap, after the setup of control paths between DEs, and leader's failure event. This mechanism fulfills the design requirements in several ways. Firstly, it does not require any pre-configuration on part of network operator beyond the DE identifiers. Secondly, it avoids the potential assignment conflicts that could arise due to asynchronous computation of assignments by DEs. Finally, it allows a robust design as failure of any particular machine does not jeopardize the network operation.

Network State and Logic: The network state, consisting of the topology information of data plane and routes advertised by DEs, is replicated across the decision plane. The route advertisements, in the form of DE-DE messages, provide reachability information about a DE's logical area. Frequent collection of topology information from the lower layers of the architecture is avoided as it is a costly process in terms of overhead and delay. This is because the abstraction of logical area boundaries does not extend to any lower layers and a request from the decision plane for collecting topology information encompasses the entire network topology. Therefore, we limit topology discovery to the cases of network bootstrap and new DE addition only. In other cases, e.g. when a DE is restarted after a failure, topology discovery is not required as it had been done previously and the persistent network state can be acquired from the current leader along with router assignments.

We categorize failure event at the data plane into, 1., Local-area events, which do not

require a re-computation of router assignments and, 2., Non-local-area events which require assignment re-computation for their resolution. Since assignment re-computation is a relatively costly network event, this distinction reduces the protocol convergence time by eliminating the need to re-compute as a result of each failure event. Local-area events result in a routing table update within the logical area where they occur, and updates from the logical area DE to other DEs in the AS. This update indicates the change in network topology and any prefixes that are unreachable as a result of router failure. On contrary, non-local-area events require a router assignment re-computation, and routing table updates in more than one logical areas. We describe the types of data plane failure events and their relation to the above-mentioned categories as follows.

- **Non-Partitioning Failure** is a router or link failure that does not result in the partitioning of the logical area where it occurred. This type of event is a local-area event. A redundant backbone router failure is an example of this failure type.
- **Disconnection Failure** results in a set of routers being unreachable from any DE in the network. The set of unreachable routers may not have failed but the loss of paths to all DEs renders them disconnected from the network. This is another case of a local-area event as re-computation of router assignments can not restore the connectivity to the disconnected set. A link failure in a router chain leading to an ISP's customer is an example of this failure type.
- **Partitioning Failure** partitions a logical area into two or more partitions, resulting in the loss of control paths to the logical area DE for at least one partition, which otherwise remains reachable from another functional and feasible DE. A router assignment re-computation restores the network control over the partition by assigning the partitioned routers to feasible DE. As a result, this type of failure is a non-local-area event.

The avoidance of router assignment re-computation in the case of local area events reduces the protocol convergence time for a significant fraction of failure events. Practical network topologies often incorporate redundancy of paths, making partitioning failures less likely. Indeed, our analysis in Section 5.4 on real-world topologies found the majority of single router failures to be local area events.

Interaction with Other 4D Layers DPP is designed to require only a small set of APIs from the underlying layers of the 4D architecture, as listed in Table 4.1. This mechanism is selected with the aim of improving extensibility of the architecture, allowing this basic set of APIs to be re-used in any additional control features beyond shortest-paths routing. The implementation of these APIs in the lower architectural layers is not explored in this work.

4.4.3 Protocol States

A DE is transitioned through several states from initialization to full operation and undergoes further state changes in response to network events. Figure 4.4 illustrates the state machine of the DPP protocol where we utilize the following states to describe its operation:

Init or initialization state follows immediately after boot-up. Secure channels for the exchange of control messages are immediately established with each of DE's neighbors in the fully-connected decision plane. If there are no previously initialized neighbors, all DEs are transitioned through the leader election protocol. Otherwise, the current leader checks a newly booted DE's identifier to find out if it was previously initialized.

Elect state is used when there is no leader DE in the network, which will be the case at network bootstrap, or in case of leader's failure. Each DE in the network is pre-configured with a unique integer identifier. The DEs exchange their identifiers to elect the one associated with the lowest identifier as leader.

Topology Discovery In this state, network topology information is requested from the 4D Discovery layer using the `get_topo()` construct. The topology is in the form of a weighted graph where vertices indicate routers and edges specify physical adjacencies, which are weighted by propagation delay of the links. The topology information is exchanged between DEs to ensure full replication of network state across the decision plane.

Router Assignment The leader DE transitions into this state in the event of a DE failure, failure at inter-area links, or an addition of a new router.

Routing Table Computation is done by each DE for the routers in its logical area whenever it receives a new assignment from leader DE, in case of intra-area events, and when it receives new reachability information from another DE. The completion of routing

table computation is immediately followed by an update of each router's routing table using the `send_RT()` construct to the 4D Dissemination plane, and an update of reachability information to other DEs if the new computation results in changes to the routes available to their logical areas.

Topology Update is a result of an event in a DE's logical area. It requires sending topology update to other DEs in decision plane in order to synchronize the network state. A `push_event()` construct allows 4D Dissemination plane to signal such events to the decision plane.

Full DE in this state indicates a fully initialized decision plane. This state would be maintained in normal operation.

4.5 NUMERICAL EVALUATION

In this section we provide results of our evaluation of the assignment algorithm on real-world and a variety of artificially generated topologies.

Rocketfuel project offers two different set of topologies that are of interest in our work with certain limitations. We utilize two sets of rocketfuel topologies. The first set is that of the backbone topologies used by Mahajan et al. [98] which is fairly restricted as it underestimates the network devices by a large margin. To compensate for this fact, we constructed a second set by parsing the data on rocketfuel's estimate of router adjacencies, matching it against the city/POP data, and using the maximal connected subgraph where the topology graph was found to be disconnected.

The second set are artificial two-tiered hierarchical topologies generated by BRITE [99] using the GLP model [100]. GLP model along with BRITE has been reported to generate ISP-like topologies [101], which we use to model a large-sized ISP topology consisting of 1500 routers and 15 DEs. We utilized two different topology models in the generation of BRITE topologies: a "large" sized ISP topology consisting of 1500 routers and 15 DEs, and a "medium" topology with 100 routers and 10 DEs. The inter-area delays in both topologies models were larger than the delays between routers in the same area. Our experiments were

repeated for different degree distribution.

The evaluation was focused on determination of the following characteristics:

1. Reassignment of non-orphaned routers: The accommodation of routers orphaned as a result of a DE failure, may necessitate re-assignment of non-orphaned routers from other DEs to balance the load among the surviving DEs. A large percentage of such reassignments could have an adverse effect on the decision plane performance and it is desirable to reduce such router churn. We measure this as a percentage of non-orphaned routers undergoing re-assignment out of the total number of routers in the network.
2. Computation time: Each failure in the decision plane triggers the re-computation of the router assignments. We measured the time taken for each run of the assignment algorithm on a 64 bit 3.6 GHz machine.

In each topology, we determine the best positioning of a set of DEs based on the discussion in Chapter 3. Results were obtained by removing all combinations of "failed" DEs from the original set. Maximum number of DEs ($n_{\max}$) was limited to 15 in BRITE and 10 in Rocketfuel sets. The minimum number of DEs $n_{\min}$ was constant at 5 in both sets, which was found to be sufficient in attaining near-optimal convergence delays[71]. The capacities of individual DEs were assumed to be a non-limiting factor and, in the case of BRITE set, our experiments were repeated for different degree distributions (d) of logical areas.

Figure 4.5 shows non-orphaned router reassignment for the case of Rocketfuel backbone topologies, where we present results by bounding the maximum percentage of router reassignments in a network and presenting the minimum value of Δ that is needed to ensure that reassignment rate remains below the bound. We observe that even in this very limiting case of backbone topologies, the rate of reassignment falls off rapidly with an increase in Δ and relatively small values of Δ are sufficient in achieving tight bounds on router reassignment. In the case of BRITE topologies, we observe even better performance as full topological information is available. Figure 4.6a and 4.6b show results for the case of BRITE topologies where we report the observed minimum values of Δ required in bounding maximum reassignments to 5% for different logical area degrees.

The computation time required to run each iteration of the algorithm is plotted in Fig-

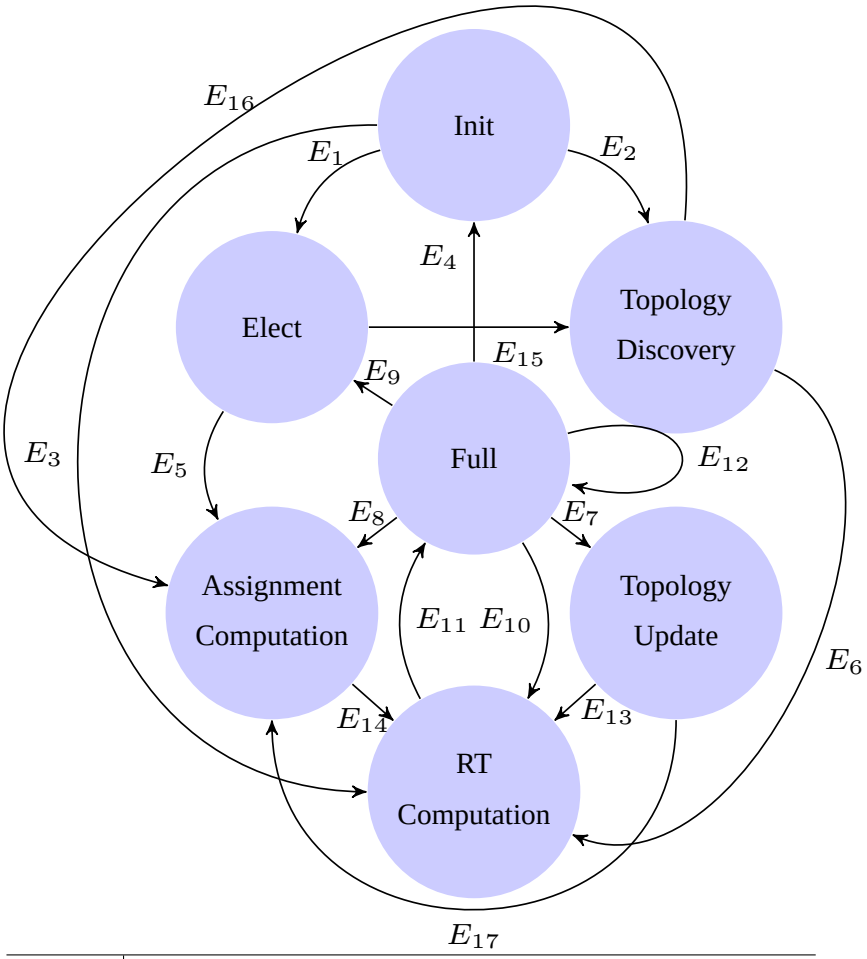
ure 4.7 for both sets of topologies, with a worst-case DE capacity constraint of $\Delta = 1.0$. The plot shows that even in case of very large network topologies and worst-case constraints on load-balancing router assignment algorithm converges to a solution within very reasonable times.

4.6 SUMMARY

This chapter presented the design of fault-tolerant logically centralized decision plane. We investigated the question of how LCDP can present a coordinated interface and maintain control over network devices in large-scale dynamically changing enterprise network. We presented an approach for adaptive maintenance of decision plane associations with the network devices. A novel and efficient algorithm for the assignment of routers to DEs was the key to our approach that optimizes the responsiveness of the decision plane and allows network operators to control the trade-off between convergence delay and load balancing across the DEs.

Furthermore, we explored the design of protocol that will allow distributed, and yet coordinated, operation of physically distributed DEs. A protocol design was presented, in the context of logically centralized route computation, that supports distributed operation and ensures replication and synchronization of network state across the logically centralized decision plane.

Finally, we presented the results of our evaluation of LCDP design over real-world and artificially generated topologies. The results show that the adaptive router assignment algorithm provides very reasonable convergence even in the case of very large network topologies and worst-case constraints on loadbalancing.



Event	***Description***
E_1	Network Bootstrap
E_2	Addition of a new DE in the network
E_3	Reception of topology and assignment from leader
E_4	Reboot
E_5	Only if not in network bootstrap
E_6	Reception of assignment from leader
E_7	Local area event
E_8	DE failure or router addition. (Leader only)
E_9	Leader failure
E_{10}	Reception of new assignment or reachability update
E_{11}	Send RTs and reachability update
E_{12}	Stable network
E_{13}	Reception of new assignment or local-area event
E_{14}	Send the assignment to other DEs in the network
E_{15}	Network Bootstrap
E_{16}	Only in the case of leader DE
E_{17}	Non local-area event (Leader only)

Figure 4.4: State transition diagram for the Decision Plane Protocol

Table 4.1: APIs used for inter-layer communication

Construct	Function
`get_topo()`	Request network topology discovery from the 4D Discovery plane.
`send_RT()`	Send a new RT to the specified router using the 4D Dissemination plane.
`push_event()`	Used by the 4D Dissemination plane to signal an event in a DE's area

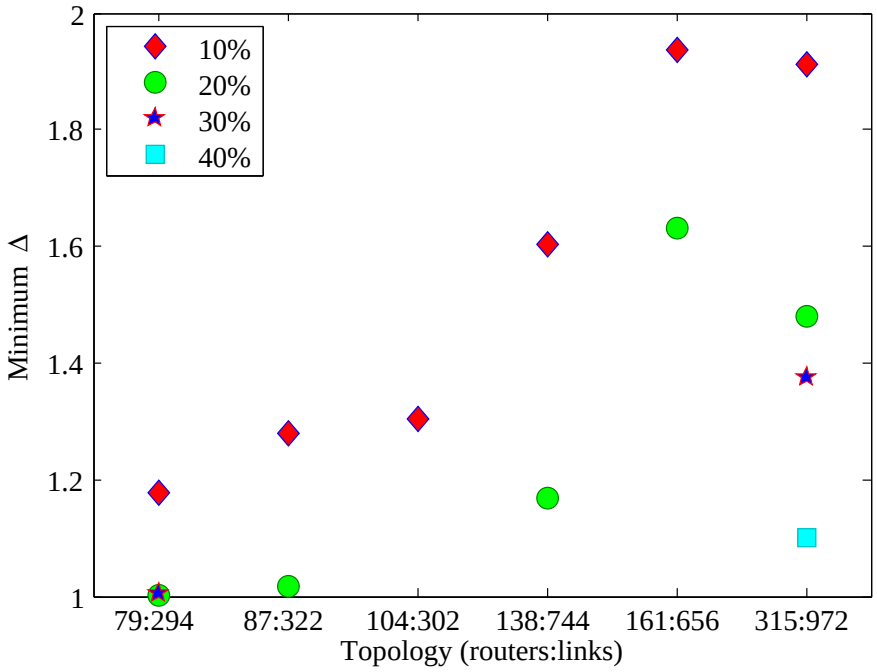


Figure 4.5: Trade-off between load balancing and percentage of on-orphaned router re-assignment for Rocketful backbone topologies. Plot shows the minimum value of Δ needed to limit the re-assignments below a given percentage.

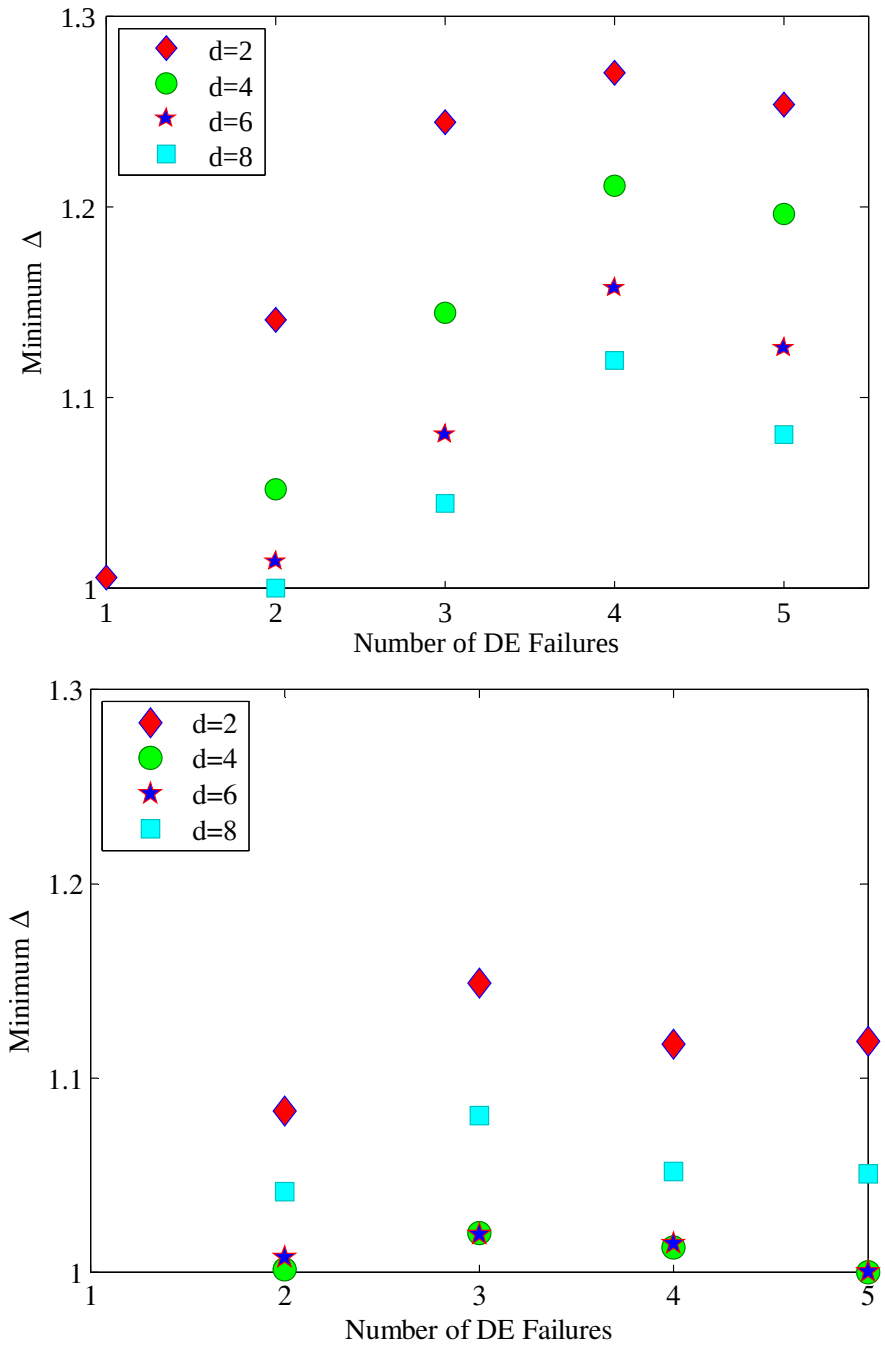


Figure 4.6: Trade-off between load balancing and percentage of on-orphaned router re-assignment for BRITE topologies. Plots show the minimum value of Δ needed to limit the re-assignments below a given percentage. Top: (a) BRITE topologies of $m = 1500$ with max. 5% re-assignments and $d_{max} = 15$, Bottom: (b) BRITE topologies of $m = 1500$ with max. 5% re-assignments and $d_{max} = 10$

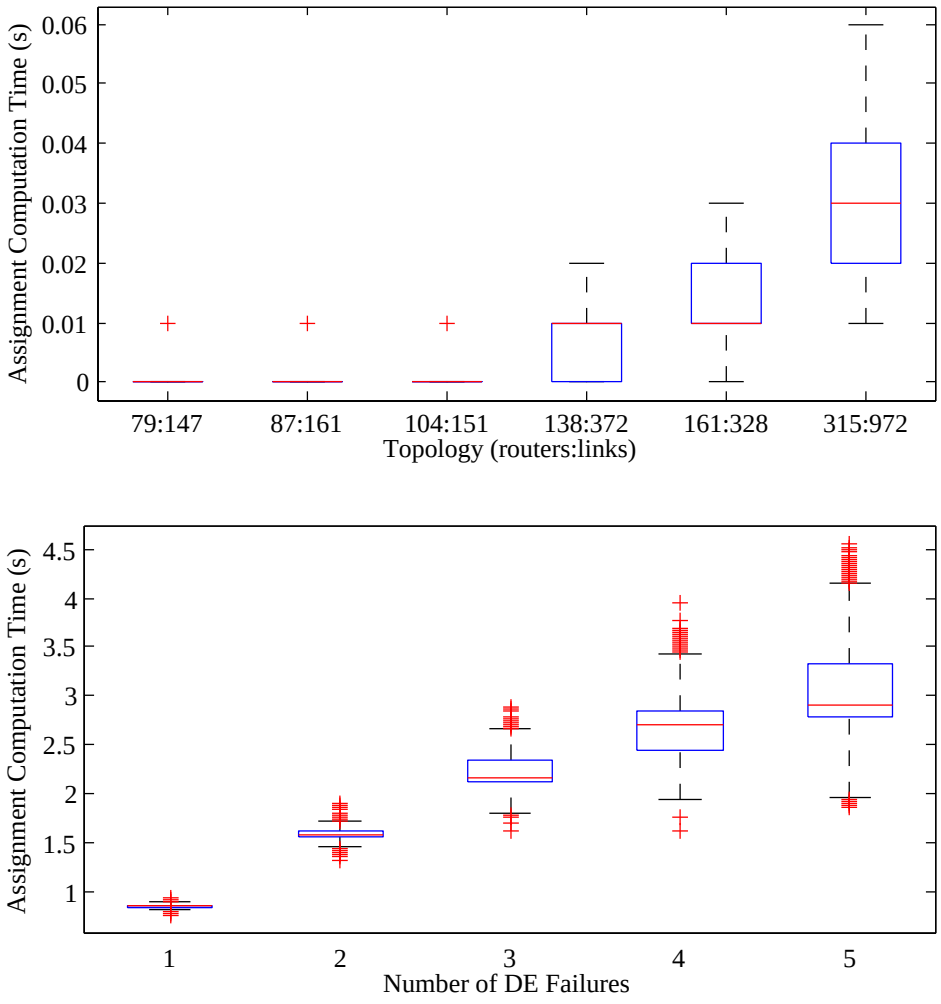


Figure 4.7: Box Plot of the computation time for router re-assignment with $\Delta = 1$. The box shows the first and third quartile along with the median. Whiskers show the min. and max. values, while the outliers are plotted as "+". Top: (a) Rocketfuel backbone topologies, Bottom: (b) BRITE topologies with $m = 1500$

5.0 TRADE-OFFS IN STATE AND DECISION-MAKING CENTRALIZATION

This chapter will present an investigation of the trade-offs involved in centralization of the state and control logic in a network. Centralization of network control state and logic provides a new and powerful approach for simplifying network management. However, it comes with some inherent trade-offs that need to be carefully considered for each control process or "application" used at the decision plane. This chapter will use Internet routing and traffic engineering as a case study for the investigation into the trade-offs.

Internet traffic engineering deals with optimally choosing the paths for traffic flows in an autonomous system. This problem is difficult to solve in a purely distributed network [102] and modern traffic engineering solutions utilize some element of centralization, such as MPLS [103]. Specifically, we will discuss the effect on decision optimality, level of state consistency, and the complexity of application logic as the network state and control logic is moved along the spectrum between physical distribution and centralization.

In the following sections an overview of the trade-offs is presented, which is followed by a discussion of the traffic engineering and optimal routing problems in current Internet design. This is followed by an discussion of how logical centralization of decision plane can be utilized to provide optimal route management in large-sized networks. A discussion of the trade-offs between centralization and distribution of state and control logic are presented next. Finally, we present simulation results of our LCDP implementation and discuss the affect on performance at varying levels of LCDP centralization.

5.1 OVERVIEW & BACKGROUND

The centralization of network control and logic in a physically or logically centralized decision plane promises to reduce the management burden on network operators and allow exploration of new opportunities for improving network management. At the same time, the delays inherent in the transport of control information, from switches to decision elements and vice versa, introduce trade-offs in maintaining consistency of network state and the appropriate level of decision plane responsiveness. The placement of control logic in the network, distribution of the state required for decision processes, and the level of state consistency required are the primary factors that control these trade-offs. The level of decision plane application complexity and desired level of performance also require additional design considerations in this context.

The design trade-offs govern a centralized decision application process's responsiveness, state consistency, complexity, and decision optimality. The physical limitation imposed by network delays and network dynamics is one of the key factors behind these trade-offs. The longer it takes to make a decision based on network traffic or event data— due to network delays and any other factor— the greater the chances are of data staleness. Often, centralization introduces a responsiveness versus degree of centralization trade-off where a higher degree of centralization might increase the response time to an event occurring in the network. Given appropriate local controls, it is possible that such an event could be handled more locally with decreased response time.

However, the trade-offs could be very different given the possible objectives for decision process. A traffic sensitive process might have a different set of trade-offs to consider from another process which considers only relatively static state information. Therefore, we observe that it is important that the network application designer carefully considers how centralization (or its relative degree) affects the process's state and logic. This analysis will define the solution space by constraining the available solutions to only those that meet the application's performance constraints. This will, in turn, derive the placement of the process's state and logic in the network.

In this chapter, an analysis of the major trade-offs in decision plane application design

is presented. This analysis is based on a case study of optimal routing as the decision plane application. The determination of optimal routing [104] paths involves finding the set of paths, given the network topology and traffic demands, that optimize an objective function. The objective function is chosen to minimize resource utilization (e.g. minimization of maximum link utilization) or maximize traffic's QoS metrics.

The key reasons for this choice is that route management is the basic decision plane process and the optimal computation of routing paths is a challenging problem that is difficult to solve using the traditional routing protocols [102]. Furthermore, this problem offers a continuum of solutions from distribution to physical centralization of control that illustrate the affect of the trade-offs. The choice of studying the trade-offs of logical centralization in the context of route management has been recently shared by Levin et al. [105]. However, while their work focused only on state distribution, this chapter will present a holistic view of the trade-off space in logical centralization, analyzing the three key trade-offs that arise with centralization of state and logic: optimality of decisions, simplicity of the protocol logic, and the level of stability in the network.

5.1.1 Traffic Engineering Model

The traffic engineering problem arises from the desire to optimally utilize the finite network resources in a network. Autonomous systems have finite network resources in the form of link capacities, packet forwarding rates and queueing capacities. Optimization of the resource utilization is an important issue for service providers as it enables optimal usage of network resources. From a practical standpoint, this allows the provider to avoid costly infrastructure upgrades that would be needed otherwise. The optimization is also beneficial from QoS prospective, as reducing resource utilization offers better network tolerance for short-term variations in traffic flows and often reduces the chances for network congestion. Consequently, traffic engineering is an important practice for large enterprises and ISP networks.

Current model of traffic engineering in Internet is generally based on optimization of the metrics used in destination based link state routing [8, 106], in conjunction with flow-based forwarding by MPLS [107, 108]. The traffic engineering process takes the network

optimization objective along with the traffic matrix, or a set thereof [109, 110, 111], as input. The TM is based on traffic history, current traffic monitoring, and contractual obligation of the organization. The output of the process gives the set of link weights that are used by the link-state routing algorithms to compute traffic paths.

The traffic engineering problem can be formulated by considering a directed graph representation of the network, $G = (V, E)$, where V is the set of nodes and E is the set of directed links. The traffic demands are assumed to be represented in the traffic matrix, D, where $D(s,t)$ denotes the traffic intensity between the source-destination pair (s,t). The load on a link, f_{ij}, is dependent on the paths chosen by the network routing policy and is upper bounded by the link capacity, c_{ij}. Furthermore, f_{ij}^t represents the flow over link ij destined for node t.

Traffic engineering problem can be modeled as a constrained multi commodity flow optimization problem, where the objective function $\phi(f_{ij}, c_{ij})$ specifies the choice of cost function used for optimization.

$$\text{Minimize } \phi(f_{ij}, c_{ij}) \tag{5.1}$$

s.t.

$$\sum_{j:(i,j)\in E} f_{ij}^t - \sum_{j:(i,j)\in E} f_{ji}^t = t(i,j) \qquad \forall i,t \in V \tag{5.2}$$

$$\sum_{t\in V} f_{ij}^t \leq c_{ij} \qquad \forall (i,j) \in E \tag{5.3}$$

$$t(i,j) = \begin{cases} -\sum_{s\in V} D(s,t) & \text{if } i = t \\ D(i,t) & \text{if } i \neq t \end{cases} \tag{5.4}$$

$$f_{ij}^t \geq 0 \tag{5.5}$$

This problem is a convex linear optimization problem if the objective function is chosen to be convex with linearly increasing cost. The choice of the exact definition of $\phi(f_{ij}, c_{ij})$ depends on the AS's routing policy. For example, minimization of maximum link utilization can be considered as an objective. In this case,

$$\phi(f_{ij}, c_{ij}) = max_{(i,j)\in E} f_{ij}/c_{ij}$$

The objective function could also be related to network performance seen by the users. For example, Fortz et al. [102] present a piecewise linear model that can be viewed as modelling retransmission delays caused by packet losses, approximating the cost function of M/M/1 queueing delay [84].

5.1.2 MPLS and Layer-2 Traffic Engineering

A common approach to Internet traffic engineering involves using layer-2 technologies or Multi-Protocol Label Switching (MPLS) [103]. Instead of using IP based traffic engineering where routing and traffic engineering decision are taken at IP layer, this approach is used to setup circuit switched paths that are not managed by layer-3 IGP route computation.

In the case of layer-2 TE, an overlay network is setup to carry IP traffic. Overlay network is created by setting up virtual paths over the physical network topology. These virtual paths form a virtual network that is opaque to the IP layer. Traffic engineering is achieved through careful optimization of traffic flows over the virtual network. The linear programming optimization problem discussed in the previous section can be used in this scenario to find the optimal traffic flows over the virtual links. There are two network management problems inherent in this TE approach. First, as the virtual links appear as physical to IP IGP protocols, the network topology from IP's perspective resembles a full mesh network. This results in scalability problems with link-state routing, as the number of router adjacencies approach the network size. In case of a router failure in such a network, each of the failed router's peers initiate a link state update. Link failures are usually even worse from scalability perspective: as physical links carry multiple virtual links, a physical link failure often appears as multiple link failures at the IP layer. This so-called "N-square" problem commonly manifests in large geographically distributed networks. The second main problem with this approach is that network operators are required to manage two different

and dissimilar data networks. The management cost of coordination between IP and layer-2 networks, including that of setting up optimal virtual paths and link metrics, can be very high.

In MPLS based TE, Label Switched Paths (LSPs) are setup to tunnel traffic aggregates between a pair of edge routers using a separate signalling protocol, such as Resource Reservation Protocol (RSVP) [108]. Since LSPs are setup over paths managed by IP routing, the scalability problem arising from full-mesh virtual topology disappears. However, the underlying network management cost of managing LSP tunnels remains comparable with layer-2 based TE. Network management needs to create and administer N^2 LSP between routers, adjust traffic loads over the LSPs, and setup alternate routes in case of local LSP failures.

5.1.3 Traffic Engineering using Inter-domain Routing Protocols

Traffic engineering using the conventional IGP protocols (OSPF, IS-IS) has an advantage of simplicity as it avoids the need for another technology for the sole purpose of TE. This simplicity translates to lesser network management costs. The basic premise of this method is to control network traffic by adjusting the link weights used in IGP route computation. As the shortest-path routing policy used in IGP prefers lesser weighted paths over higher weighted ones, a carefully selected set of link weights can be used to direct traffic along optimal paths.

However, this model of traffic engineering has several obvious drawbacks. First, there is no provision of directly influencing the decision taken by the routing protocols. As a result, protocol decision are influenced by AS-wide optimization of link weights. This process of finding the correct values of link weights for TE optimization is known to be NP-hard [102]. Second, the distributed routing algorithms used in inter-domain routing forward traffic only along the shortest path(s) from a source to a destination. There is no provision for forwarding traffic along paths that may not be shortest in terms of link weights, but can help in increasing the overall network utility, such as by being used for alleviating congestion on the shortest path.

5.1.4 Adaptive Routing

The adaptive routing paradigm differs from off-line TE by allowing the routing policy to automatically change paths based on the prevalent traffic conditions. In this model, the routing protocol monitors the traffic load on links and uses this information in its routing decision, for example by diverting traffic away from a congested path.

The main differences between this scheme and general TE methods discussed earlier are time scale of change and automatic behavior of adaptive routing. Traffic traversing any link can change rapidly in its magnitude due to short term variations in traffic demands or as a result of failures of network equipment. Traffic engineering is mostly concerned with optimizing routing for longer term averages and frequent route optimization is avoided as it may impact the entire network's traffic flows and can have large period of transient behavior where routing loop may occur. Adaptive routing, on the other hand, is concerned with optimizing routing paths even over short periods of time. The time scale over which adaptive routing re-computes the optimum routing paths is therefore smaller than that of a traffic engineering process. Furthermore, as routing adaptation is part of a routing protocol, the process runs without needing any input from a human operator.

The main disadvantage of adaptive routing is often its oscillating behavior and poor stability that can occur in networks with changing traffic conditions and traffic loads approaching link capacities [112]. The timescale of change in traffic demands is often short in comparison with the transient period of a routing change. This creates a problem as the routing process needs to continuously monitor for traffic conditions that trigger a re-computation of routing paths. Without any reliable mechanism to enforce instantaneous routing change across the network, routers update their routing tables with the newly computed paths. This process may take a significant amount of time during which routing loop may occur due to inconsistencies in routing paths at different routers. This entire process of abnormal traffic load detection, route re-computation, and post-routing change transient needs to occur at a timescale shorter than the rate of change in network traffic. Otherwise, routing paths will remain unstable if the rate of change in traffic is shorter than the time routing process needs to re-converge to a stable state.

5.2 TRAFFIC ENGINEERING WITH LOGICALLY CENTRALIZED DECISION PLANE

As discussed in the previous sections, traffic engineering is inherently a centralized network management task. A decision plane offering centralized control and management service could become an ideal point for the implementation of this function. There are several characteristics of LCDP design that can be useful for this application.

- **TE Centralization** offers global views of the AS network state facilitating easier real-time data collection for generating traffic matrices. The dissemination plane provides the signaling glue between data plane devices (routers, switches, etc.) and decision plane and allows direct traffic data collection. Implementation of data collection mechanisms can take the form of either a timer based scheme, where traffic information is sent to the decision plane periodically, or a "trap" based scheme where traffic load going beyond a set threshold triggers a message to the decision plane. Either way, the centralization of logic that can solicit, store, and analyze the traffic information and use it to optimize the traffic paths offers a powerful new platform for traffic engineering. Furthermore, the utility of centralization is notably significant for TE, as algorithms for generating optimal routing paths such as multi-commodity network flow optimization naturally lead to centralized solutions [113].
- **Multipath Forwarding** The decision plane paradigm is not constrained by backward compatibility with conventional route computation, such as seen in OSPF based TE. Unlike current IGP based TE solutions, this design does not restrict multipath forwarding and allows more tractable and robust solutions that are not dependent on current design choices. Multipath forwarding is generally constrained to equal-cost multipaths in today's IGP protocols [114].
- **Scalability** of current schemes is constrained by the NP-hard optimization of OSPF's link weights [102] or by the difficulty in management of MPLS paths [115]. Both of these constraints disappear in a traffic engineering approach based on clean-slate decision plane.
- **Adaptive & Online TE** Unlike current IGP based solutions, a decision plane based TE design is not restricted to offline traffic engineering and allows for an automated real-time

traffic engineering scheme. Offline schemes such as OSPF-TE [106] optimize the network traffic for long-term demands and the resulting solution can be suboptimal due to short-term traffic dynamics in the network. Robustness to unpredictable network failures is also suboptimal in offline TE schemes as their fault-tolerance is dependent on proactive pre-computing of alternate paths for different failure scenarios. In a large network with thousands of devices and links, the number of all the potential failure vectors is far greater than what can be feasibly considered in pre-computed optimization. This leads to schemes that provide TE solutions that remain workable under a large number of failure scenarios, at the cost of optimality under any particular failure [116]. Decision plane based TE does not have these design shortcomings as a decision plane can make traffic engineering decision online, and thereby recompute the traffic paths in response to failures or changes in the network.

Figure 5.1 presents the vision of a traffic engineering framework based on a logically centralized decision plane. As the figure shows, a decision plane houses the TE logic, collects traffic measurements, gathers topology information. Optimal traffic routes are computed at the decision plane in accordance with the TE policy set by the network manager and communicated to the network devices.

There are several design trade-offs that need to be taken into account in a scheme that supports TE under a logically centralized decision plane paradigm. As opposed to a physically centralized design where one physical device, a single Decision Element, is responsible for network-wide control and management, a logically centralized design offers more location possibilities where traffic engineering decisions can be taken. Placement of TE logic at each DE can lead to a more distributed and potentially more robust design than a scheme where TE decision are taken on a single DE in a network. However, there are several factors here that need consideration. First, routing decisions taken by TE logic need to be consistent over the entire network, otherwise routing loops can occur. This consistency of routing decisions, in turn, demands synchronization in network state across a geographically distributed set of DEs — a difficult requirement to meet due to distributed nature of the design. On the other hand, our objectives of a scalable and robust design favor a distributed approach where local decision-making at each DE for the area under its control can lead to shorter response time

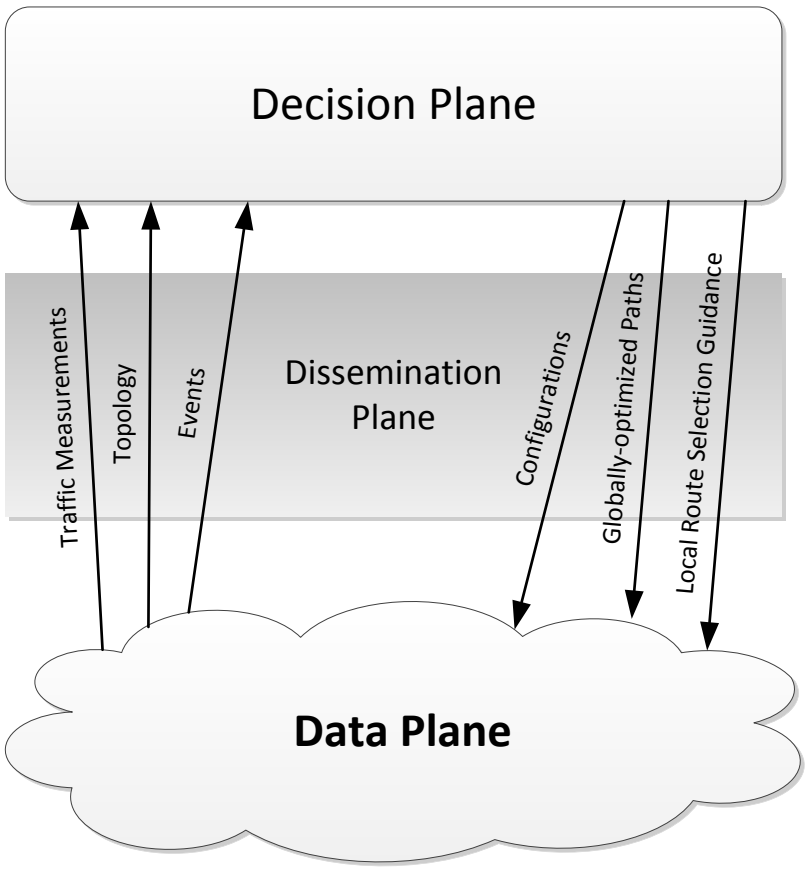


Figure 5.1: Traffic engineering in a logically centralized decision plane

of the decision plane and generally a more scalable design. Relative strictness of solution optimality provides another trade-off as strictly optimal TE favors a more centralized approach, as opposed to more flexibility in a design where sub-optimal TE solutions can be tolerated.

There are several operational considerations that affect the placement trade-off of TE decision logic. Factors such as network size, geographical dispersion, bounds on convergence delay, and the desired level of solution optimality affect the trade-off in placement of TE logic. A relatively large network size, for example, results in longer convergence delays between a failure and convergence to stable state in a design where TE decision logic is concentrated at a single location. On the other hand, this design choice may be workable in a small network,

depending on the bounds on convergence dictated by management policy. This suggests that a TE design needs to be flexible enough to allow its applicability with a diversity of different network types and under different performance requirements.

5.3 TRADE-OFFS IN LOGICALLY CENTRALIZED TRAFFIC ENGINEERING

This section discusses the different trade-offs an application designer faces in designing an application for logically centralized decision plane. We identify and discuss the three key trade-offs that arise between the global optimality of decisions, stability of application operation, and the simplicity of the control logic. These trade-offs arise due to the choices that are afforded by LCDP in distributing the application's state and control logic on the spectrum between physical distribution and physical centralization.

The distribution, or the level thereof, of state that is required by the application is one of the key factors governing the trade-offs. We consider the application model where the actual network state is desired as the input to the control processes. This model works well for the common network processes such as route management, but may be to be relaxed for application scenarios where network state is either not needed or is considered to be static. For those applications that depend on the collection of network state, the key issue with centralization is the level of inconsistency that arises when a centralized DE or controller collects state information from network nodes that may be physically far from itself. The distance between the controller and network nodes increases the staleness of controller's viewpoint at any time. This affects the decisions taken by the controller in two ways — first, the controller's collected network state is susceptible to staleness, affecting the decision optimality; and second, as the controller's decisions need to be received by network nodes before they are implemented, state distribution adds delay between the time a decision a taken by the controller and when the nodes receive and implement it.

Similarly, the control logic that derives application decisions can be distributed to the level of network nodes, on one end of the spectrum, and centralized at the decision plane on

the other end. This level of control distribution is coupled with the distribution of state, but the coupling is not always strict. Also, the control logic can be distributed unevenly in the network between the centralized controllers and the network nodes, giving varying levels of autonomous control to the network nodes.

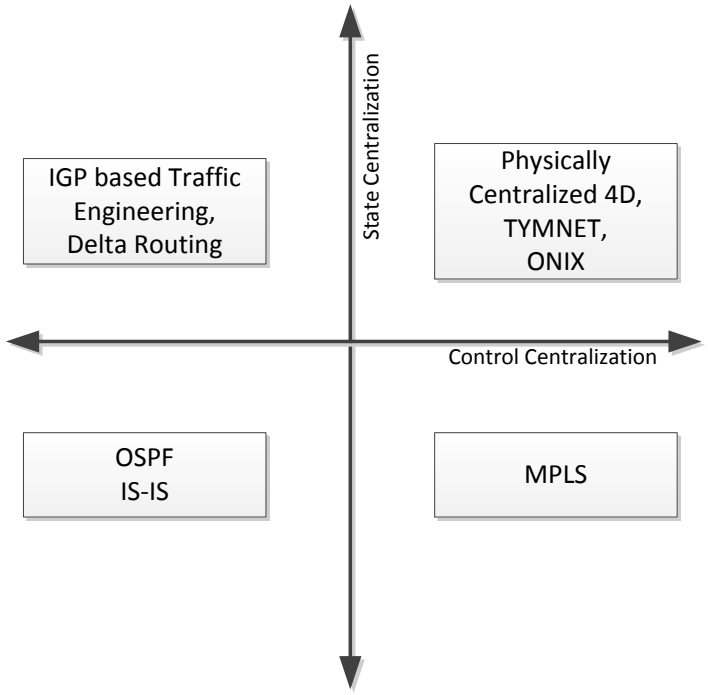


Figure 5.2: State and control logic distribution in routing algorithms

Figure 5.2 provides several examples of networking protocols that show the choices between centralization and distribution of state and control logic. In this chart, centralization of both control logic and state information is exemplified by physically centralized 4D architecture along with the ONIX architecture [69] and legacy Tymnet [45]. In each of these designs, the centralized controller is responsible for collecting the network-wide state information, using the centralized protocol control logic to make decisions on routing and other applications, and relaying the decisions to the network nodes. On the other side of the spectrum, traditional routing protocols, such as OSPF [117] and ISIS [22] shown in the lower left quadrant of the chart, exemplify the choice of distributing both control and state in the network. MPLS [107] and ATM [47] provide examples of centralized control logic and distributed state with their similar approaches to path and connection setup. Finally, the

upper left quadrant shows examples of centralized state information and distributed control logic with IGP-based Traffic engineering, discussed earlier in this chapter, and Delta routing [118] where individual nodes carry control logic to choose paths within the limitations specified by the centralized controller. These examples show that there are many choices that an application designer can make in choosing the appropriate level of distribution of control logic and state information for any particular decision plane application.

5.3.1 Trade-offs

The level of distribution of control logic and state information, along with the choices of algorithms used in the application decision processes, define the landscape of decision plane trade-offs for an application designer. The choices made for these govern the three key factors in logically centralized application design:

1. **Optimality** of the decisions taken by the decision plane, where we assume that the application designer seeks to optimize the application decisions to increase the application utility.
2. **Stability** of the application operation, where we informally define stability as characterised by lack of any "transient" application state in the network. This covers the transient period of time after a state change in the network necessitates an application to recompute and the time until the application decisions are fully implemented in the network.
3. **Simplicity** of the logic used by the application, where simplicity can be defined both in terms of the algorithmic complexity of problem or the need to employ sophisticated techniques to infer or predict network state in the absence of actual (or measured) state information.

These three trade-offs can be viewed as being jointly dependent where an increase in utility of any two of them comes at the expense of the third. We motivate the discussion of this observation by an example of network traffic engineering using LCDP in Figure 5.3a. In this example network, the (logically) centralized DE is responsible for computing the paths taken by network flows based on the topology and flow demands collected from the network

nodes. Here a simplistic traffic engineering application might utilize a centralized approach of optimizing a traffic cost function based on flow demands and other constraints collected from the network. The use of minimum cost multicommodity flow optimization [113] is one example of this approach. While this approach will provide solution optimality, the trade-off with network stability is possible due to the state inconsistency between DE and the network nodes, which is a consequence of the latency between the nodes and the DE. As the network size grows larger, the delay in both updating the DE of any network events and disseminating the DE decisions to the nodes will grow. This delay will directly impact the stability of the network as it will be become difficult to maintain a consistent network-wide view at the decision plane or synchronize state change across the data plane devices.

Figure 5.3b depicts the approach where the responsiveness of the decision plane is improved. Trading off application simplicity by employing either an "Oracle" DE, with an ability to gather instantaneous network state information, or through the use of pre-computed routes can improve both the stability and optimality of the application. In practical terms, the oracle DE could employ techniques to predict network information, in this case the flow information, based on past history of network state. On the other hand, pre-computed routes can be associated with triggers in network flow state such that nodes can independently measure network traffic and utilize appropriate routes as instructed by the DE in advance. This technique will be similar to the one used by Delta routing [118]. Both oracle-based or pre-computed state techniques can incur significant cost to the simplicity of application logic relative to the case of centralized cost optimization. Here, the application's controlling logic is distributed, increasing its complexity, to gain more flexibility in synchronizing application's state.

The third case of trade-off, depicted in Figure 5.3c, happens when optimality of an application is traded for an increase in application stability and simplicity. An example of this approach is the setup of paths that are less susceptible to changes in network state. This can be done by using a Pareto optimal solution over a set of expected traffic matrices [109].

5.3.2 Analysis for Traffic Engineering Application Design

Let $R = \{r_1, r_2, ..., r_i\}$ be the collection of routers and switches in a network area which is under the control of a Decision Element (DE). This area can be the entirety of the network if physical centralization of decision plane is used or could be one of the partitions of the network that are each under the control of a logically centralized DE. We denote the propagation delays between the router r_i and the controlling DE as t_{p_i} and let $T_p = max_i(t_{p_i})$.

We assume the model of Figure 5.4 for route computation. This model aims to maintain network state consistency at the decision plane and provides opportunity for choosing appropriate trade-off between fully centralized decision-making and varying levels of decentralization. The levels can range from pre-computed routes at the routers to provisioning of local decision logic at the routers. After a state change occurs in the network, the model assumes that there will be period of time before an application's control logic starts path recomputation. This will cover the time taken to detect the event, either at the level of individual routers for events such as link failures, or the level of the decision plane for route recomputation due to divergence of collected traffic measurements from the optimal solution. In any case, this period of time is assumed to involve components such as the various hold-down timers used in OSPF and IS-IS, along with the transient period of time required to process routing changes such as modifying state at the line cards based on routing updates. We denote these transient time intervals as T_t and note that the actual propagation time from routers to DE is not a component of T_t. Furthermore, we assume that T_c is the time taken by the decision plane logic to process the network state and output the new routes.

In case of centralized decision making, it is now possible to characterize the total *service* time of the network event as $T_s = 2T_p + T_t + T_c$, where $2T_p$ is the component introduced by network delays of collecting state information at the DE and disseminating the routes from DE to the routers. In networks with multiple hops and where significant congestion delay can be expected in the paths between routers and DEs, an additional component to capture queueing delays might be appropriate.

Under the assumption of a Poisson arrival process, with arrival rate of λ, for the network events that result in recomputation of routes, we model the processing of events and route

computation as an M/M/1 queue. The stability condition of such systems requires that $\lambda < \mu$, where μ is the service rate of the system. In the case of centralized routing decisions, this results in

$$\lambda < \frac{1}{T_s} = \frac{1}{2T_p + T_t + T_c} \tag{5.6}$$

Equation 5.6 suggests that there is an inverse relationship between the responsiveness of the decision plane to changes in network state and the delay in maintaining consistent state at the decision plane. For large sized networks, where remote routers can be situated far from the DEs, the network propagation delays will affect the response time of the decision plane. Consequently, the decision plane will only be able to respond to infrequent state changes, with a smaller value of λ.

We now consider that case where decision plane is able to decentralize the control of the network by placing pre-computed routes at the routers along with state triggers that will invoke the transition from one set of routes to another. This hybrid approach [118, 83] is a cross between fully centralized and fully distributed decisions and places some of the control logic inside the routers to enable faster response to network events. In this case, assuming that a fraction α of route changes can be pre-computed at a computation cost of T_c' based on network state, we get

$$\lambda < \frac{1}{(1-\alpha)(2T_p + T_t + T_c') + \alpha T_t} \tag{5.7}$$

Here λT_t is assumed to be the service time of events where the pre-computed routes were used. This component of the total service time is considered to be much less that the service time needed for DE based computation as no network or computational delays are involved.

Equation 5.7 suggests that the responsiveness of decision plane can be increased if the control logic can be decentralized efficiently. However, this is dependent on the feasibility of computing routing paths in advance of their actual need and the need to balance the trade-off with increased computational cost and protocol complexity.

To summarize, the propagation delays in large network topologies restrict the level of decision plane responsiveness to network state changes. One way of improving responsiveness is to compromise on the desired level of application optimality. The other mechanism, which aligns well with the design philosophy of LCDP, is to decentralize the control logic and bring control decisions closer to the data plane devices. This could be achieved by placing a subset of control logic at the routers, e.g. by the use of pre-computed routes. This would allow the logically centralized decision plane to provide the network-wide coordination and route guidance without incurring the cost in reduced responsiveness. However, as the discussion of tradeoffs in the previous section suggests, this approach could result in higher complexity of application logic.

5.4 SIMULATIVE EVALUATION OF LCDP DESIGN

This section presents the simulation results of our investigation of LCDP's performance, especially in the context of application design tradeoffs.

We analyzed the convergence performance of LCDP with simulations on Rocketfuel ISP topologies, using ns-2 simulator [1] where we created new modules to implement the functionality of LCDP. We collected results on the convergence delay in cases of network bootstrap, DE, and router failures.

Convergence delays in the case of DE failures were computed by randomly forcing the failure of a DE and measuring the time until all routers in the network receive re-computed routing tables. This convergence delay includes: delay at the decision plane between the time a failure actually occurs and when it is detected by the functional DEs; computation time of router assignment algorithm; reception of new assignments by the DEs; new routing table computation; and, reception of new routing tables at each router.

The decision plane failures are detected by a DE keep-alive timer which expires when no keep-alive message is received by a neighboring DE within a time period equal to the maximum delay between DEs. We utilized results obtained in the previous chapter for routing as-

[1] http://isi.edu/nsnam/ns/

signment computation time while routing table computation time was kept constant at 1ms. Simulation were repeated for the range of DE failure combinations with $n_{\max} = 10,\ n_{\min} = 3$.

Table 5.1 shows convergence and maximum network delays for network bootstrap, that is typically higher than normal operation as network adjacencies are collected from scratch. Box plot of the convergence delays in case of DE failure, under normal operation, is shown in Figure 5.5. The figure shows that even with the large scale simulated topologies, our design achieves sub-second convergence delays. This result is similar to the reported performance of optimized conventional intra-domain routing [119] and shows that the LCDP design is able the network management benefits of logical centralization while matching the widely accepted performance requirements of the Internet.

Convergence in case of router failures is simulated by randomly stopping a router instance in ns-2 and measuring the time for protocol convergence. After the detection of failure at the logical area DE, a hold-down timer of $30ms$ is used to detect correlated failures. Upon the expiry of hold-down timer, the DE checks the type of the event and requests a re-assignment from the leader in case of non local-area events. Shortest path route computation at the DEs was handled by Floyd-Warshall algorithm. The router assignments in the simulated topologies were computed using $\Delta = 1$ and $n_{\max} = 10$. We simulated single and multiple router failure cases separately.

Figure 5.6 shows the boxplots of protocol convergence delays after single router failures for 100 random failures in each Rocketfuel topology. The results are shown for the cases of 2, 5, and 10 DEs. We can notice that, for most of the topologies, the convergence performance of network with 5 DEs is significantly better than with only 2 DEs. Also, results with 10 DEs don't show significant improvement over the ones with 5 DEs. This observation is in line with the results from Chapter 3 where diminishing returns were seen for networks with more than 5 decision elements.

Figure 5.7 shows similar results of protocol convergence delays after multiple router failures for 100 random failures in each Rocketfuel topology. As expected, the convergence performance in this case is generally worse than single failure results. However, for most of the topologies the difference in performance is not significantly different.

We found that no more than 8% of the router failure were non local-area events. Since

only non local-area events require assignment re-computation, the vast majority of router failure events did not require router assignment re-computation. In failure cases where assignment re-computation was required, the convergence delays were similar to DE failure results of Figure 5.5. This is the reason for the similarity in performance between single and multiple router failure results. Even though there were more failures in the latter case, the DEs were able to recompute routes for their areas without recomputing the assignments. This result also reinforces the observation made in the previous section that an approach of handling events locally can improve the responsiveness and performance of centralized decision planes and should be pursued if the associated increase in application complexity can be efficiently handled.

Finally, Figure 5.8 shows the difference in average convergence performance for the router failures that resulted in router reassignment computation, and those that did not. The figure shows that router reassignment computation resulted in an increase of around 20% over the simulated topologies.

The results show that the techniques described in this dissertation for logically centralzied decision plane design can achieve sub-second convergence delays even in largest of the simulated topologies for both router and DE failure cases. This convergence delay performance compares favorably with the reported studies of optimized IGP convergence in conventional distributed routing protocols [119].

5.5 SUMMARY

This chapter discussed the trade-offs a decision plane application designer faces in striking the right balance of application state and logic distribution in a logically centralized network. Using the problem of network traffic engineering as a decision plane application example, we explored the application design space and discussed the three key trade-offs that arise in logical centralization, namely between optimality of an application's decisions, stability of its operation, and the relative simplicity of the logic governing application's operation. In the context of traffic engineering, we demonstrated through examples that an application de-

signer is likely to compromise on one of the trade-off factors to increase the utility function of the other two factors. We also observed that stability and responsiveness of the logically centralized networks can be increased if some of the application's control logic is distributed to the network's data plane nodes. This can enable faster response to network events while still maintaining global coordination of decisions through guidance from centralized component of application logic.

Finally, we presented the simulation results of our investigation of LCDP design using large-scale real world topologies. The results show that proposed design of logically centralized decision plane achieves sub-second convergence delays even in largest of the simulated topologies for both router and DE failure cases. This convergence delay performance compares favorably with the reported studies of optimized IGP convergence in conventional distributed routing. Furthermore, the results reinforced the observations made earlier in this chapter about the existence of the three application design tradeoffs. The results also show the value in placing distributed control logic closer to the network data plane for increasing the responsiveness of logically centralized design.

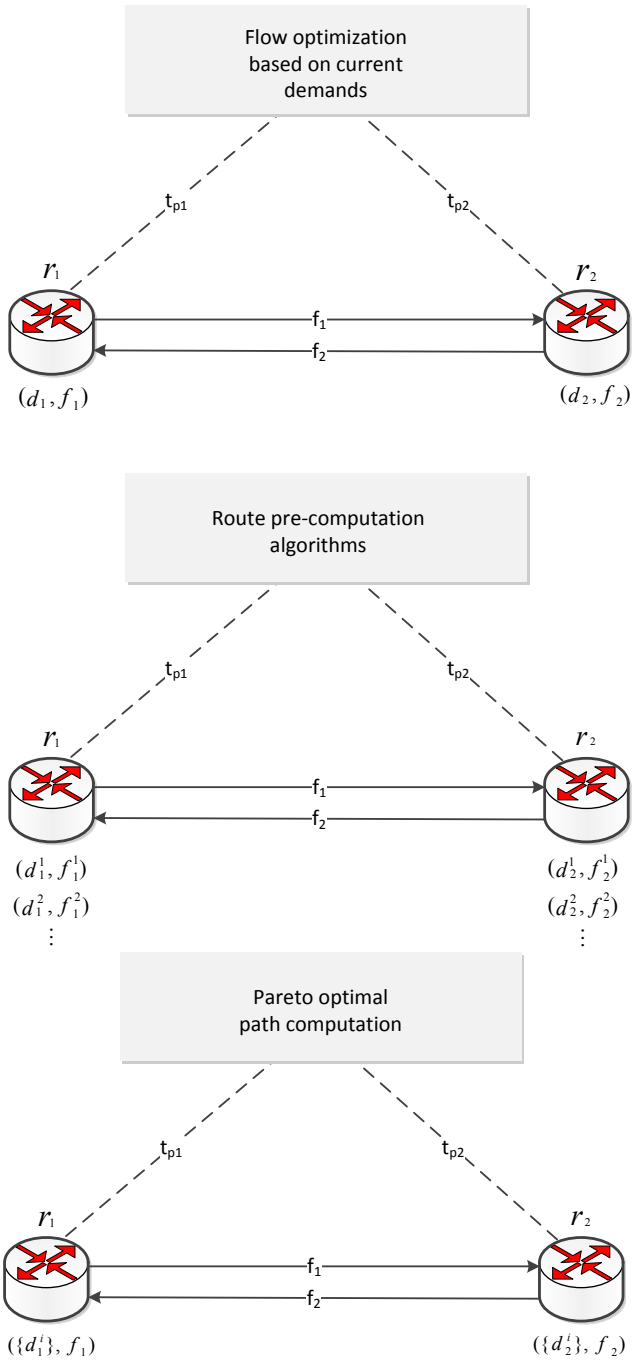


Figure 5.3: Examples of trade-offs in path optimality, application simplicity, and stability in traffic engineering context. Two routers, r_1 & r_2, are shown with their flow demands and assignments. Top: (a) A centralized solution with application stability trade-off. State synchronization between decision and data plane is limited by network delays, Middle: (b) Application simplicity trade-off with the use of pre-computed flows. Positioning some of the control logic at data plane reduces state synchronization constraints. Bottom: (c) Decision optimality trade-off with the use of Pareto-optimal paths over a set of traffic demands

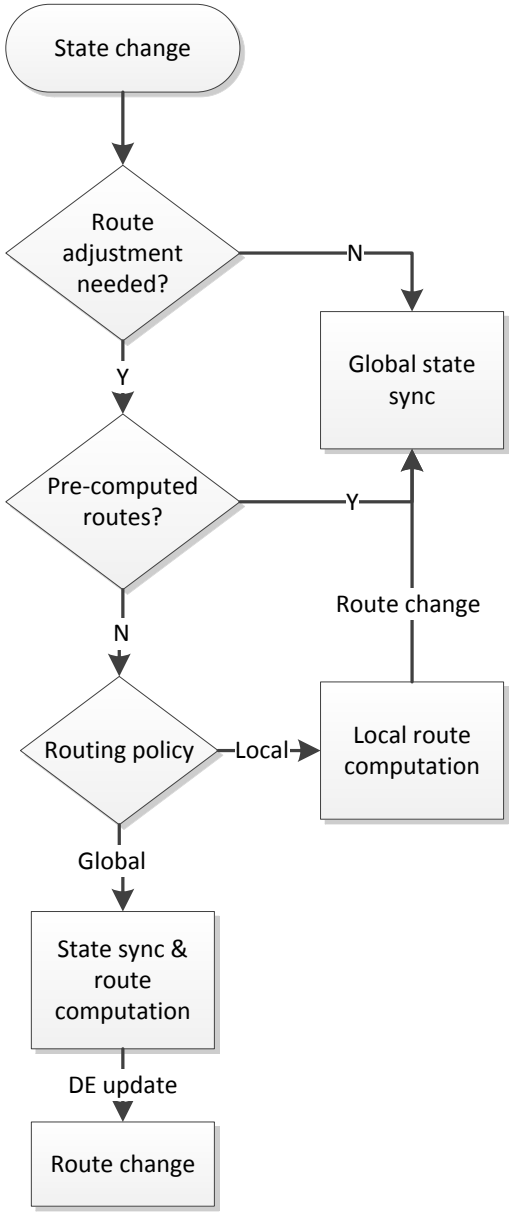


Figure 5.4: Route computation model

Table 5.1: Bootstrap convergence delays for Rocketfuel topologies

Topology (routers:links)	**Max. Network Delay** (ms)	**Bootstrap Delay** (ms)
104:151	28	95.13
87:161	35	126.35
161:328	47	175.12
79:147	72	235.3
317:972	86	306.4
138:372	97	383.2

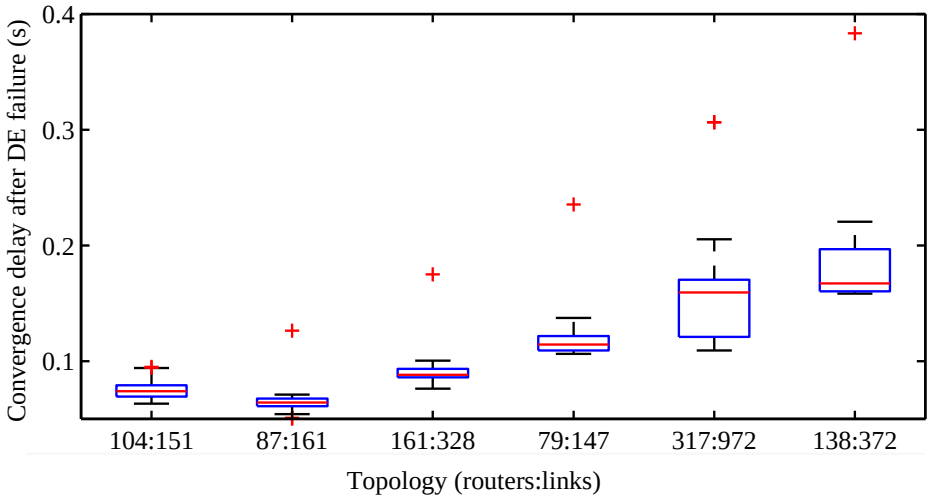


Figure 5.5: Box Plot of protocol convergence delay after DE failures for Rocketfuel topologies with $n_{\max} = 10$ and $\Delta = 1$. The box shows the first and third quartile along with the median. Whiskers show the min. and max. values, while the outliers are plotted as "+".

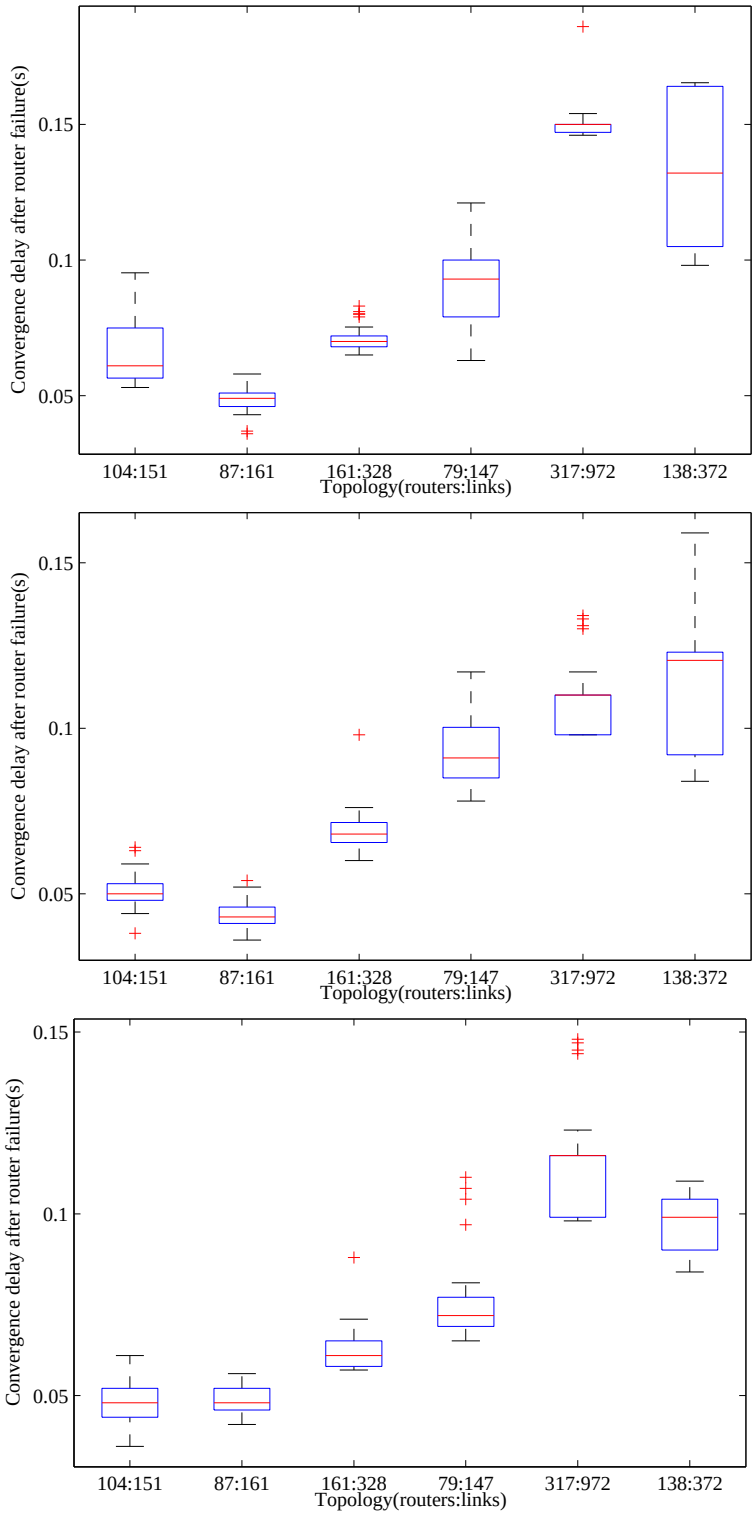


Figure 5.6: Box Plot of the observed convergence delays after single router failure in Rocketfuel topologies. The box shows the first and third quartile along with the median. Whiskers show the min. and max. values, while the outliers are plotted as "+". Top: (a) LCDP with 2 DEs, Middle: (b) LCDP with 5 DEs, Bottom: (c) LCDP with 10 DEs

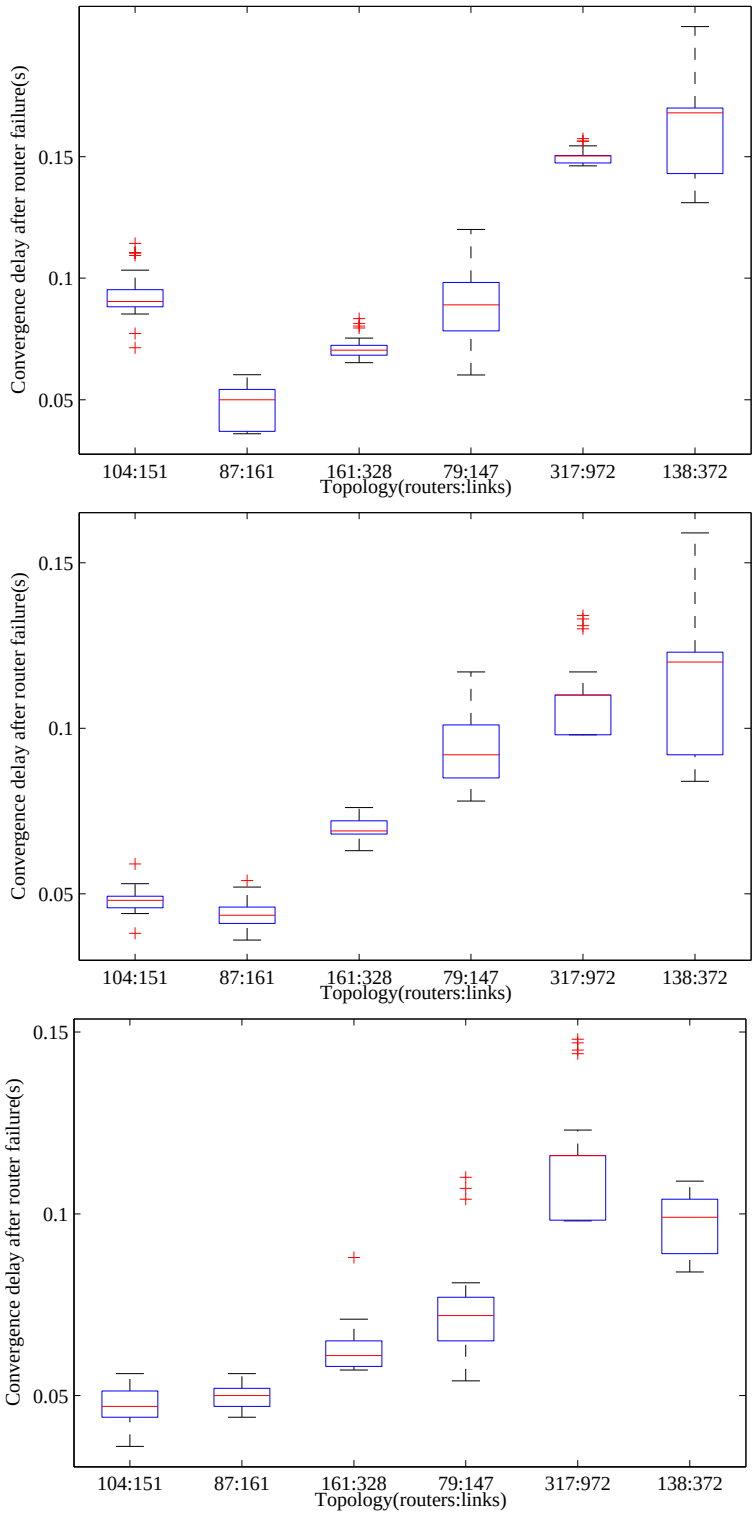


Figure 5.7: Box Plot of the observed convergence delays after multiple router failures in Rocketfuel topologies. The box shows the first and third quartile along with the median. Whiskers show the min. and max. values, while the outliers are plotted as "+". Top: (a) LCDP with 2 DEs, Middle: (b) LCDP with 5 DEs, Bottom: (c) LCDP with 10 DEs

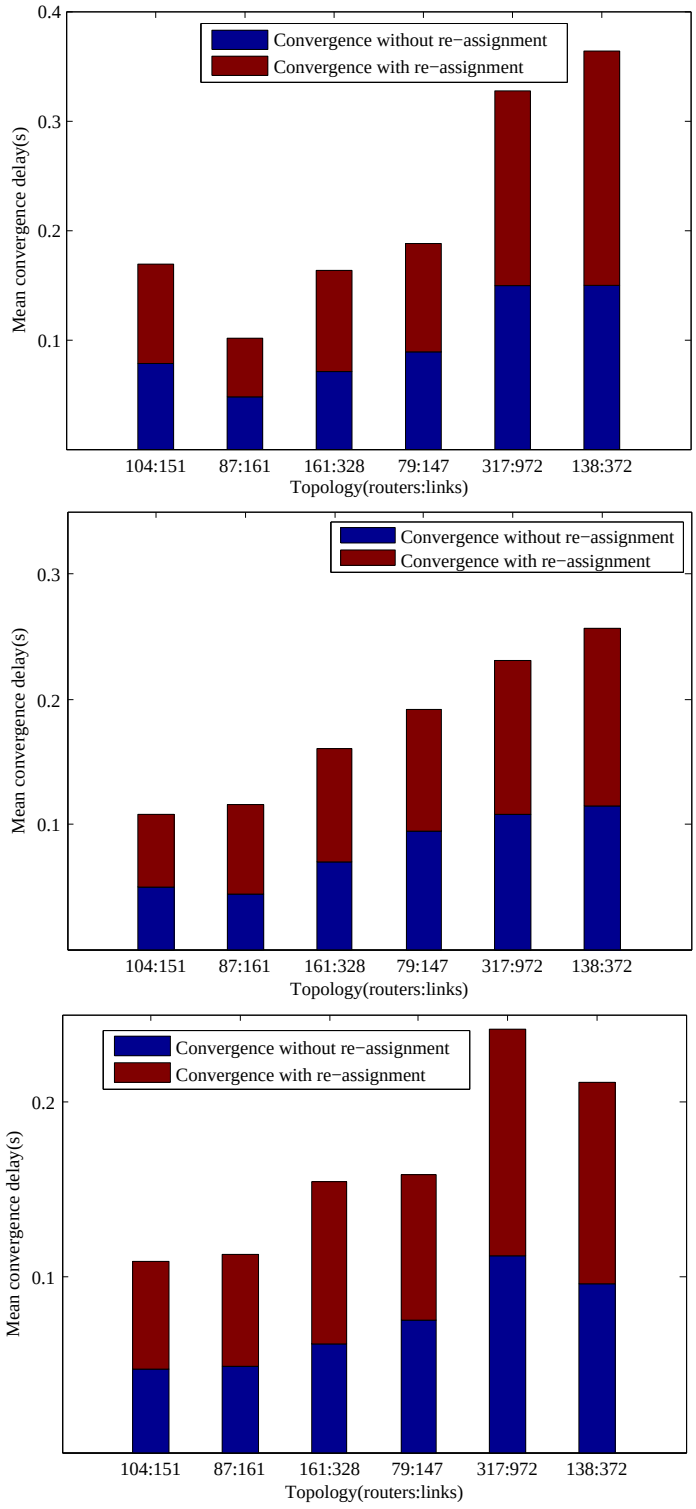


Figure 5.8: Plot of the mean convergence delays with, and without, re-assignments for Rocketfuel topologies. Top: (a) LCDP with 2 DEs, Middle: (b) LCDP with 5 DEs, Bottom: (c) LCDP with 10 DEs

6.0 CONCLUSION AND FUTURE WORK

This dissertation presented an investigation into the novel networking construct of a *logically centralized and physically distributed* decision plane for managing large-scale computer networks. We argued that both complete distribution, or centralization, of control in a network constrains the task of network management, either by introducing unnecessary complexity in the former case, or by limiting the scalability and robustness of network control in the latter. We presented a balanced design based on logical centralization which permits a set of network controllers to collaboratively govern network devices. This design allows greater flexibility over centralized control architectures to network infrastructure and application designers in balancing the need for consistent network control with timely control over network events.

We discussed the need for optimizing the physical design of a logically centralized decision plane and provided algorithms that would help a network architect choose the appropriate positioning of decision elements in the network. The results of our investigation over real world network topologies indicate that, even in large sized ISP topologies, it is possible to position a relatively small number of decision elements in a way that maximizes the responsiveness of decision plane. We also investigated the feasibility of a dynamic logically centralized decision plane which adapts to changes in network topology and re-configures itself to ensure optimal decision plane responsiveness. This lead to our design of an algorithm for optimal router assignment and a protocol for decision plane operation. Using our dynamic router assignment algorithm and protocol, we measured the convergence of a route computation application over ISP topologies and found that the proposed scheme provides sub-second convergence properties that compare favorably with those of current IGP routing protocols.

Finally, we investigated the tradeoffs that are found in application design space of logically centralized and physically distributed decision-making using traffic engineering as an example application. This investigation led to the conclusion that there are three distinct application design factors that determine the placement of application state and logic in a network, namely the desired optimality of application decisions, stability of its operation, and simplicity of application logic. These factors are interdependent and need to be jointly optimized according to the needs of an application. We argue that adjustment of these tradeoffs allows an application designer to find the right balance between the extremes of physical centralization and distribution by customizing the placement of a control application's state and logic according to objectives and constraints of an application.

In conclusion, this thesis argued that logical centralization provides a feasible design alternative to the distributed nature of conventional network protocols or the physical centralization offered by some of the proposed architectures for network re-design. The key contribution of the thesis is in the demonstration of how this framework can be practically adopted for large enterprise and ISP networks where network management complexity is currently a major management concern.

6.1 FUTURE WORK

Future work in the area of logical network centralization can take several exciting directions related to the contributions of this dissertation. One of these directions is in further development of the LCDP paradigm towards its practical realization in networks. The second direction is along the line of application design for leveraging the newly offered capabilities.

6.1.1 Protocols and Algorithms for Decision Plane Operation

Decision plane protocol can be improved with more detailed specification of protocol operation and extended analysis of its characteristics in a variety of network settings. The simulative analysis of DPP protocol can be improved by the addition of a variety of failure

scenarios such as multiple correlated failures, failures at the physical layer, and a detailed DE failure model that considers different component failures. The impact of router failures on protocol stability can also be analyzed beyond the simple failures scenarios considered here, perhaps by incorporating additional network topologies and failure models.

Extension of our work in route assignment can consider a variety of optimization metrics and optimization problems that are relevant to different network settings, going beyond provisioning of basic reachability. Furthermore, a challenging problem exists in the detailed analysis of DE capacities and the impact of non-homogeneous work load.

Other main avenues of related future research include the interoperable design of the decision plane with lower layers of the architecture and extension of management functions, e.g. to include provisions for system maintenance.

6.1.2 Deployment Strategies and Legacy Infrastructure Support

One key concern with clean-slate Internet re-architecture proposals has been that, without support for backward compatibility, their practical realization often means replacing existing infrastructure. This viewpoint is valid for academic research as it helps us understand the architectural alternatives but it is fairly impractical in the real world to assume that legacy infrastructure could be replaced in the foreseeable future to accommodate new architectures.

The logically centralized decision plane presented in this thesis was conceived as a clean slate design but is envisioned to coexist with the legacy Internet infrastructure. There are two main future research avenues that can help in bringing this to practical reality: research on decision plane applications that can coexist with the traditional routing protocols to enable logically centralized and legacy heterogeneous network segments in an autonomous system; and applications that can allow re-purposing of legacy network devices for use inside the network segments controlled by LCDP.

6.1.3 Application Development

Some of the most exciting future research opportunities lies in the development of decision plane applications that can leverage the possibilities offered by logical centralization. We

argue that in addition to the basic network functions of providing and managing reachability, new applications can cover the following areas.

- **User Mobility** presents a difficult challenge in the traditional network design that is largely based on assumption of static end hosts. With the prevalence of mobile devices and growth of adhoc networks, it is becoming increasingly important for the Internet to seamlessly handle user mobility within, or across, autonomous systems. The centralization offered by the decision plane presents an opportunity to integrate route and mobility management within an AS, and coordinate mobility decisions with other decision planes and management systems in the global Internet.
- **Internet Security** is an area where the centralized decision plane can be especially effective by enforcing network-wide policies, ensuring routing congruence with access control, and participating in active monitoring of the data plane. There are many avenues of research in this direction that also have been a focus of several ongoing and related efforts. An exciting opportunity exists in network user and traffic policy enforcement where logical centralization can help in joint optimization of network authentication and route management. A centralized decision plane also provides a very attractive location for network defense measures including a role in network-wide correlation of threat signatures and formulation of appropriate response.
- **Energy Efficiency** in enterprise and service provider network is another area where logically centralized route computation can be useful. The combination of network-wide visibility and ability to affect the traffic paths at the decision plane could be leveraged for optimizing routes based on energy efficiency constraints. Future research could investigate the feasibility of directing traffic intelligently to selected data centers in a large network based on traffic and energy patterns.

BIBLIOGRAPHY

[1] D. Clark, "The design philosophy of the darpa internet protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 106–114, Aug. 1988. [Online]. Available: http://doi.acm.org/10.1145/52325.52336

[2] R. Braden, D. Clark, S. Shenker, and J. Wroclawski, "Developing a Next-Generation Internet Architecture," 2000, ISI White Paper. [Online]. Available: http://www.isi.edu/newarch/DOCUMENTS/WhitePaper.ps

[3] World internet usage statistics news and population stats. [Online]. Available: www.internetworldstats.com/stats.htm

[4] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4d approach to network control and management," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, 2005.

[5] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford, "The cutting EDGE of IP router configuration," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 21–26, January 2004.

[6] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding bgp misconfiguration," in *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4. New York, NY, USA: ACM, Aug. 2002, pp. 3–16. [Online]. Available: http://doi.acm.org/10.1145/964725.633027

[7] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.

[8] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional ip routing protocols," *Communications Magazine, IEEE*, vol. 40, no. 10, pp. 118 – 124, oct 2002.

[9] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. V. D. Merwe, "The case for separating routing from routers," in *Proc. of FDNA '04. ACM SIGCOMM Workshop on Future Directions in Network Architecture.* ACM Press, 2004.

[10] D. A. Maltz, G. Xie, J. Zhan, and H. Zhang, "Routing design in operational networks: A look from the inside," in *Proc. of ACM SIGCOMM '04*. ACM Press, 2004, pp. 27–40.

[11] A. Wool, "A quantitative study of firewall configuration errors," *Computer*, vol. 37, no. 6, pp. 62 – 67, june 2004.

[12] R. Ramjee, F. Ansari, M. Havemann, T. V. Lakshman, T. Nandagopal, K. K. Sabnani, and T. Y. C. Woo., "Separating control software from routers." in *Proc. of COMSWARE '06. International Conference on Communication System Software and Middleware*, 2006.

[13] A. Greenberg, G. Hjalmtysson, D. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "Refactoring network control and management: A case for the 4D architecture," Carnegie Mellon University, Tech. Rep. CMU-CS-05-117, Sept 2005.

[14] H. Yan, D. Maltz, T. Ng, H. Gogineni, H. Zhang, and Z.Cai, "Tesseract: A 4D network control plane." in *Proc. of NSDI '07. Symposium on Networked Systems Design and Implementation*. USENIX, April 2007.

[15] R. Braden, "Requirements for Internet Hosts - Communication Layers," RFC 1122 (Standard), Internet Engineering Task Force, Oct. 1989, updated by RFCs 1349, 4379, 5884. [Online]. Available: http://www.ietf.org/rfc/rfc1122.txt

[16] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*. Addison-Wesley, 2010. [Online]. Available: http://books.google.com/books?id=gxLePQAACAAJ

[17] J. Hawkinson and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)," RFC 1930 (Best Current Practice), Internet Engineering Task Force, Mar. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1930.txt

[18] D. A. Maltz, G. Xie, J. Zhan, H. Zhang, G. Hjálmtýsson, and A. Greenberg, "Routing design in operational networks: a look from the inside," in *Proc. of SIGCOMM '04*. ACM, 2004.

[19] G. Malkin, "RIP Version 2," RFC 2453 (Standard), Internet Engineering Task Force, Nov. 1998, updated by RFC 4822. [Online]. Available: http://www.ietf.org/rfc/rfc2453.txt

[20] B. Albrightson, J. Garcia-Luna-Aceves, and J. Boyle, "EIGRP - A Fast Routing Protocol Based On Distance Vectors," in *Proc. Networld/Interop 94*, 1994.

[21] J. Moy, "OSPF Version 2," RFC 2328 (Standard), Internet Engineering Task Force, Apr. 1998, updated by RFC 5709. [Online]. Available: http://www.ietf.org/rfc/rfc2328.txt

[22] D. Oran, "OSI IS-IS Intra-domain Routing Protocol," RFC 1142 (Informational), Internet Engineering Task Force, Feb. 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1142.txt

[23] I. Stoica, "Stateless Core: A Scalable Approach for Quality of Service in the Internet," CARNEGIE MELLON UNIVERSITY, Tech. Rep., 2000.

[24] F. Baker, "Requirements for IP Version 4 Routers," RFC 1812 (Proposed Standard), Internet Engineering Task Force, Jun. 1995, updated by RFC 2644. [Online]. Available: http://www.ietf.org/rfc/rfc1812.txt

[25] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. Mckeown, "Buffer sizing in all-optical packet switches," in *In Proceedings of OFC/NFOEC*, 2006, pp. 5–10.

[26] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2000, pp. 1435 –1444 vol.3.

[27] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework," RFC 3410 (Informational), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3410.txt

[28] W. Stallings, *SNMP,SNMPV2,Snmpv3,and RMON 1 and 2*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[29] M. T. Rose, *The Simple Book: An Introduction to Networking Management: Revised Second Edition*, 2nd ed. Simon & Schuster Trade, 1995.

[30] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. USA: Addison-Wesley Publishing Company, 2009.

[31] W. Enck, P. McDaniel, S. Sen, P. Sebos, S. Spoerel, A. Greenberg, S. Rao, and W. Aiello, "Configuration management at massive scale: system design and experience," in *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2007, pp. 6:1–6:14.

[32] J. Gottlieb, A. Greenberg, J. Rexford, and J. Wang, "Automated Provisioning of BGP Customers," *IEEE Network*, vol. 17, pp. 44–55, 2003.

[33] CISCO Inc., "CISCO Configuration Engine." [Online]. Available: http://www.cisco.com/en/US/products/sw/netmgtsw/ps4617/index.html

[34] Juniper Networks, "Provider Network Management." [Online]. Available: http://www.juniper.net/us/en/solutions/service-provider/network-service-management/

[35] Hewlett-Packard Development Company, "HP Network Automation software."

[36] EMC Corporation, "EMC Ionix Network Configuration Manager." [Online]. Available: http://www.voyence.com

[37] CPlane Inc., "CPLANE InSight."

[38] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411 (Standard), Internet Engineering Task Force, Dec. 2002, updated by RFCs 5343, 5590. [Online]. Available: http://www.ietf.org/rfc/rfc3411.txt

[39] Cariden Inc., "Cariden MATE Framework." [Online]. Available: http://www.cariden.com/products/

[40] OPNET Inc., "OPNET SP Guru Network Planner." [Online]. Available: http://www.opnet.com/solutions/network_planning_operations/spguru_network_planner

[41] Arbor Networks Inc., "PeakFlow SP." [Online]. Available: http://www.arbornetworks.com/peakflowsp

[42] I. T. Union, "ITU-T recommendation Q.700: Introduction to CCITT Signalling System No. 7," 1993.

[43] R. Thompson, *Telephone Switching Systems*. Artech House, 2000.

[44] D. Lynch, J. Gray, and E. Rabinovitch, Eds., *SNA and TCP/IP Enterprise Networking*. Prentice Hall, 1997.

[45] L. Tymes, "Routing and flow control in TYMNET," *IEEE Transactions on Communications*, vol. 29, pp. 392–398, Apr 1981.

[46] H. Zimmermann, "OSI reference model–the ISO model of architecture for open systems interconnection," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 425–432, Apr 1980.

[47] W. Stallings, *High Speed Networks and Internets: Performance and Quality of Service*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[48] C. Hedrick, "Routing Information Protocol," RFC 1058 (Historic), Internet Engineering Task Force, Jun. 1988, updated by RFCs 1388, 1723. [Online]. Available: http://www.ietf.org/rfc/rfc1058.txt

[49] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771 (Draft Standard), Internet Engineering Task Force, Mar. 1995, obsoleted by RFC 4271. [Online]. Available: http://www.ietf.org/rfc/rfc1771.txt

[50] B. Carpenter and S. Brim, "Middleboxes: Taxonomy and Issues," RFC 3234 (Informational), Internet Engineering Task Force, Feb. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3234.txt

[51] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford, "The cutting EDGE of IP router configuration," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 21–26, 2004.

[52] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang, "Impact of configuration errors on dns robustness," in *Proc. of SIGCOMM '04*. ACM, 2004.

[53] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031 (Proposed Standard), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3031.txt

[54] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. der Merwe, "Design and implementation of a routing control platform," in *Proc. of NSDI '05. Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2005.

[55] J. V. der Merwe et al., "Dynamic connectivity management with an intelligent route service controlpoint," in *SIGCOMM workshop on Internet Network Management (INM)*. ACM, 2006.

[56] T. V. Lakshman, T. Nandagopal, R. Ramjee, K. Sabnani, and T. Woo, "The SoftRouter architecture," in *Proc. of ACM HotNets-III workshop*. ACM, 2004.

[57] G. Hjlmtyss, "The pronto platform - a flexible toolkit for programming networks using a commodity operating system," in *OpenArch*, 2000.

[58] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, pp. 217–231, 1999.

[59] Y. Wang, I. Avramopoulos, and J. Rexford, "Morpheus: making routing programmable," in *Proc. of INM '07. 2007 SIGCOMM workshop on Internet network management*. New York, NY, USA: ACM, 2007, pp. 285–286.

[60] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: live router migration as a network-management primitive," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 231–242.

[61] M. Agrawal, S. R. Bailey, A. Greenberg, J. Pastor, P. Sebos, S. Seshan, K. van der Merwe, and J. Yates, "RouterFarm: towards a dynamic, manageable network edge," in *Proceedings of the 2006 SIGCOMM workshop on Internet network management*, ser. INM '06. New York, NY, USA: ACM, 2006, pp. 5–10.

[62] E. Keller, J. Rexford, and J. Van Der Merwe, "Seamless BGP migration with router grafting," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, ser. NSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 16–16.

[63] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "Sane: a protection architecture for enterprise networks," in *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. Berkeley, CA, USA: USENIX Association, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267346

[64] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.

[65] H. Ballani and P. Francis, "CONMan: a step towards network manageability," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 205–216, 2007.

[66] Z. Cai, F. Dinu, J. Zheng, A. L. Cox, and T. S. E. Ng, "The preliminary design and implementation of the maestro network control platform," Rice University, Tech. Rep. TR08-13, Oct 2008.

[67] Z. Cai, "Maestro: Achieving scalability and coordination in centralized network control plane," Ph.D. dissertation, Rice University, 2011.

[68] (2012). [Online]. Available: http://www.openflow.org/wp/openflow-components/

[69] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*, 2010, pp. 351–364.

[70] A. Tootoonchian and Y. Ganjali, "Hyperflow: a distributed control plane for openflow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, ser. INM/WREN'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 3–3. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863133.1863136

[71] H. Iqbal and T. Znati, "Distributed control plane for 4D architecture," in *Proc. of GLOBECOM '07. IEEE Global Telecommunications Conference.* IEEE, Nov. 2007, pp. 1901–1905.

[72] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 2, pp. 5–17, 1996.

[73] S. R. Srinivasan, J. W. Lee, E. Liu, M. Kester, H. Schulzrinne, V. Hilt, S. Seetharaman, and A. Khan, "NetServ: dynamically deploying in-network services," in *Proceedings of the 2009 workshop on Re-architecting the internet*, ser. ReArch '09. New York, NY, USA: ACM, 2009, pp. 37–42.

[74] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: role-based architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 17–22, Jan. 2003. [Online]. Available: http://doi.acm.org/10.1145/774763.774765

[75] J. Koehler, C. Giblin, D. Gantenbein, and R. Hauser, "On autonomic computing architectures," IBM Research, Tech. Rep. RZ 3487, 2003. [Online]. Available: http://www.zurich.ibm.com/pdf/ebizz/idd-ac.pdf

[76] [Online]. Available: http://www.ana-project.org/

[77] R. Rastogi, Y. Breitbart, M. Garofalakis, and A. Kumar, "Optimal configuration of ospf aggregates," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 2, pp. 181 – 194, apr 2003.

[78] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks performance evaluation and optimization," *Computer Networks (1976)*, vol. 1, no. 3, pp. 155 – 174, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0376507577900022

[79] A. Basu and J. Riecke, "Stability issues in ospf routing," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '01. New York, NY, USA: ACM, 2001, pp. 225–236. [Online]. Available: http://doi.acm.org/10.1145/383059.383077

[80] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 251–262, Aug. 1999. [Online]. Available: http://doi.acm.org/10.1145/316194.316229

[81] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in internet topologies," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 2, pp. 18–28, Apr. 2000. [Online]. Available: http://doi.acm.org/10.1145/505680.505683

[82] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *Proc. of SIGCOMM '02*. ACM, 2002, pp. 133–145.

[83] K.-W. Kwong, L. Gao, R. Guérin, and Z.-L. Zhang, "On the feasibility and efficacy of protection routing in ip networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1543–1556, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1109/TNET.2011.2123916

[84] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1992.

[85] D. Clark, J. Wroclawski, K. Sollins, and R. Braden, "Tussle in cyberspace: defining tomorrow's internet," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 462–475, 2005.

[86] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley-Intersci., 1995.

[87] I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search," *International Transactions in Operational Research*, vol. 1, no. 3, pp. 317–336, 1994. [Online]. Available: http://dx.doi.org/10.1111/1475-3995.d01-43

[88] L. A. Lorena and E. L. Senne, "A column generation approach to capacitated p-median problems," *Computers & Operations Research*, vol. 31, no. 6, pp. 863 – 876, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030505480300039X

[89] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. pp. 699–719, 1966. [Online]. Available: http://www.jstor.org/stable/168733

[90] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations Research*, vol. 46, no. 3, pp. pp. 316–329, 1998. [Online]. Available: http://www.jstor.org/stable/222825

[91] J. Kelley, J. E., "The cutting-plane method for solving convex programs," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. pp. 703–712, 1960. [Online]. Available: http://www.jstor.org/stable/2099058

[92] P. Frana, N. M. Sosa, and V. Pureza, "An adaptive tabu search algorithm for the capacitated clustering problem," *International Transactions in Operational Research*, vol. 6, no. 6, pp. 665–678, 1999. [Online]. Available: http://dx.doi.org/10.1111/j.1475-3995.1999.tb00180.x

[93] V. Maniezzo, A. Mingozzi, and R. Baldacci, "A bionomic approach to the capacitated p-median problem," *Journal of Heuristics*, vol. 4, no. 3, pp. 263–280, Sep. 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1009665717611

[94] *GLPK (GNU Linear Programming Kit) Reference Manual*, Free Software Foundation, Inc, 2006.

[95] L. C. Freeman, "A set of measures of centrality based on betweenness." *Sociometry*, vol. 40, pp. 35–41, 1977.

[96] A. Kershenbaum, *Telecomm. Network Design Algorithms.* McGraw-Hill, Inc., 1993.

[97] S. Khuri and T. Chiu, "Heuristic algorithms for the terminal assignment problem," in *In Proceedings of the 1997 ACM Symposium on Applied Computing.* ACM Press, 1997, pp. 247–251.

[98] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proc. of INM '02. ACM SIGCOMM Internet Measurement Workshop.* ACM, 2002, pp. 231–236.

[99] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," in *MASCOTS*, 2001.

[100] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," in *Proc. of INFOCOM '02*. IEEE, 2002.

[101] O. Heckmann, M. Piringer, J. Schmitt, and R. Steinmetz, "On realistic network topologies for simulation," in *Proc. of MoMeTools '03. ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*. ACM, 2003.

[102] B. Fortz, "Internet traffic engineering by optimizing ospf weights," in *in Proc. IEEE INFOCOM*, 2000, pp. 519–528.

[103] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," RFC 2702 (Informational), Internet Engineering Task Force, Sep. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2702.txt

[104] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *Commun. Surveys Tuts.*, vol. 10, no. 1, pp. 36–56, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1109/COMST.2008.4483669

[105] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized?: state distribution trade-offs in software defined networks," in *Proceedings of the first workshop on Hot topics in software defined networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342443

[106] B. Fortz and M. Thorup, "Optimizing ospf/is-is weights in a changing world," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 4, pp. 756 –767, May 2002.

[107] X. Xiao, A. Hannan, and B. Bailey, "Traffic engineering with mpls in the internet," *IEEE Network Magazine*, vol. 14, pp. 28–33, 2000.

[108] D. Awduche, "MPLS and traffic engineering in IP networks," *Communications Magazine, IEEE*, vol. 37, no. 12, pp. 42 –47, Dec. 1999.

[109] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley, "On optimal routing with multiple traffic matrices," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, march 2005, pp. 607 – 618 vol. 1.

[110] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 248–258.

[111] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: methodology and experience," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 265–280, June 2001.

[112] Z. Wang and J. Crowcroft, "Analysis of shortest-path routing algorithms in a dynamic network environment," *ACM Computer Communication Review*, vol. 22, pp. 63–71, 1992.

[113] D. G. Cantor and M. Gerla, "Optimal routing in a packet-switched computer network," *IEEE Trans. Comput.*, vol. 23, no. 10, pp. 1062–1069, Oct. 1974. [Online]. Available: http://dx.doi.org/10.1109/T-C.1974.223806

[114] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992 (Informational), Internet Engineering Task Force, Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2992.txt

[115] E. Osborne and A. Simha, *Traffic Engineering with MPLS*. Pearson Education, 2002.

[116] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05, 2005, pp. 253–264.

[117] J. Moy, "OSPF Version 2," RFC 1247 (Draft Standard), Internet Engineering Task Force, Jul. 1991, obsoleted by RFC 1583, updated by RFC 1349. [Online]. Available: http://www.ietf.org/rfc/rfc1247.txt

[118] H. Rudin, "On Routing and "Delta Routing": A Taxonomy and Performance Comparison of Techniques for Packet-Switched Networks," *IEEE Transactions on Communications*, vol. 24, pp. 43–59, 1976.

[119] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 35–44, 2005.