Traffic Engineering for Hybrid Optical and Electronic Switching Networks

by

Richard Rizkallah Rabbat
B.E., American University of Beirut (1994)
M.E., American University of Beirut (1996)
S.M., Massachusetts Institute of Technology (1998)

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the field of Communication Networks
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Civil and Environmental Engineering
May 4, 2001

Certified by .....
Kai-Yeung (Sunny) Siu
Associate Professor of Mechanical Engineering
Thesis Supervisor

Certified by .....
Steven R. Lerman
Professor of Civil and Environmental Engineering
Chairperson, Doctoral Thesis Committee

Accepted by .....
Oral Buyukozturk
Chairman, Departmental Committee on Graduate Studies

Traffic Engineering for Hybrid Optical and Electronic Switching Networks

by

Richard Rizkallah Rabbat

Submitted to the Department of Civil and Environmental Engineering on May 4, 2001, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of Communication Networks.

Abstract

Quality of Service (QoS) over the Internet is receiving increasing attention with growing need to support upcoming multimedia applications. Many of these applications including real-time video and audio traffic require a more robust architecture that can deliver faster response times to the service requested. The Internet infrastructure currently supports a best effort service paradigm that does not differentiate between different flows. To support future applications, this thesis proposes an approach to solve the QoS needs of the traffic the network carries, by reserving bandwidth, reducing delay and increasing availability. The issues addressed in this dissertation are two-fold, leading to a better network switching architecture to support the differing needs of high-priority and low-priority voice and data traffic.

Link failure is a problem that seriously affects QoS-enabled routing. The thesis addresses this challenge by designing a mechanism to restore network connectivity and reach optimality in the event of failures, while using a variant of link-state routing protocols.

The thesis applies insights from the first problem to design an improved switching/routing architecture that services the needs of both low-priority and high-priority traffic. It achieves this architecture by making intelligent traffic admission and transport and assigning that traffic to packet switching or circuit switching hardware, in this case, an IP router and an all-optical cross-connect combined in a single hybrid switch design.

Thesis Supervisor: Kai-Yeung (Sunny) Siu
Title: Associate Professor of Mechanical Engineering

Thesis Reader: Steven R. Lerman
Title: Professor of Civil and Environmental Engineering

Acknowledgements

I would first and foremost like to thank my advisor Sunny Siu, for his encouragement, constant guidance during the research and the writing of this thesis. Sunny did the best a student could ask from his advisor. He made the years that I spent at MIT enjoyable and a great learning experience. He helped me in both my technical skills as well as my writing and documentation skills.

I would like to thank Steve Lerman, my committee chairman, for his unfettered support during my stay at the Center for Educational Computing Initiatives, as well as his great feedback at all committee meetings. It was under his leadership that I was able to grow intellectually at MIT.

Kevin Amaratunga, committee member as well, gave me a lot of advice on putting better focus in my work and enhancing my presentation skills, which helped me deliver several motivational discussions to communicate properly what I was trying to solve, and ultimately understand better the problems I was solving.

I would also like the opportunity to thank Cynthia Stewart and Jessie Williamson who have helped my doctoral research tremendously by providing me with advice, helping me with scheduling all these different milestones that I went through to deliver this piece of work and checking the correctness of my final copy with various formatting requirements.

Several of my family members have moved to the Boston area or have been visiting quite regularly. My cousins in the United States, Canada, Belgium, South Africa, Saudi Arabia, Spain, Ireland and Lebanon are have been always been tons of fun and plenty encouraging.

Let me also mention friends, Lebanese, American and international. You made my stay at MIT entertaining and shared your energy, wit, jokes, etc. I would like to mention my lab mates here and gone Paolo, David, Anthony, Edmond, Ching, Thit and Mingxi. For that I thank you all and I'd like to mention also: Saad, Walid, Issam, Ahmad, JC, Joe, Lisa, Maria, my lifelong friend, Alicia the Danish architect and Madeline, the MBA who worked with me on writing an award-winning business plan based on this thesis.

The Lebanese Club at MIT was a great experience for me. It helped me understand that my point of view was not the correct one all the time. I learnt things about my country that I had never known because I was on "the other side". The frequent and sometimes heated discussions that I had with Issam Lakkis, Ibrahim Abou-Faycal and my roommate Saadeddine Mneimneh were eye-opening experiences that gave me a great deal of knowledge. Florence Eid was my partner-in-crime in organizing a lot of lectures with speakers either Lebanese or with interest in Lebanon. Her work as a faculty member at the American University of Beirut is having a great effect on the recognition of Beirut as a place of opportunities for the venture capital world.

I would also like to especially mention my grandmothers, Labibeh and the memory of Helene. Labibeh is the model of a person who can strive in an otherwise harsh environment. Her energy throughout the war in Lebanon was a model for me in bad times and helped me understand that excuses did not help and that only hard work and optimism delivered the. Helene was the most welcoming person I had ever known and she taught me good hospitality and how to have an open heart.

It has been about four years that I have not been back to my country, Lebanon. Although I miss it quite a bit, I also feel torn between my allegiance to my hometown

Beirut and the new town that has embraced me and helped me thrive: Boston. Beirut and Boston will always represent what I like and dislike most in the world. They carry in their hearts the best and the worst days of my life and for that, I love you both.

I would like to close on all this personal outpouring of sentiments by quoting an author, poet, philosopher and artist genius that I admire most, who shared my love for Lebanon and Boston, and to whom multiple generations are forever endowed. In the words of Gibran Khalil Gibran [32], my farewells to MIT and the great learning that I had here.

> Then said a teacher, "Speak to us of Teaching."
>
> And he said:
>
> No man can reveal to you aught but that which already lies half asleep in the dawning of our knowledge.
>
> The teacher who walks in the shadow of the temple, among his followers, gives not of his wisdom but rather of his faith and his lovingness.
>
> If he is indeed wise he does not bid you enter the house of wisdom, but rather leads you to the threshold of your own mind.
>
> The astronomer may speak to you of his understanding of space, but he cannot give you his understanding.
>
> The musician may sing to you of the rhythm, which is in all space, but he cannot give you the ear, which arrests the rhythm nor the voice that echoes it.
>
> And he who is versed in the science of numbers can tell of the regions of

weight and measure, but he cannot conduct you thither.

For the vision of one man lends not its wings to another man.

And even as each one of you stands alone in God's knowledge, so must each one of you be alone in his knowledge of God and in his understanding of the earth.

To Jody, Josephine, Ronald, Ralph

I LOVE YOU!

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Market Changes of the Last Few Years

A few years ago, the explosion of the Internet came about. The Mosaic web browser of the NCSA transformed into Netscape® Communicator™, the software that brought about the revolution that is still causing businesses to look at new ways of developing and deploying their strategies. The Telecommunications Act of 1996 that deregulated the telecommunications industry was the main catalytic event for this explosion. File transfers became more complicated, and data become richer and more complex. Whereas a few years ago, most of the Internet traffic consisted of static web viewing, multimedia content started to slowly emerge as more and more companies developed their Internet presence, providing live web casts of political events, concerts and conferences. IP telephony traffic is growing as more and more people use their Personal

Computers to communicate over the world. On the other hand, crucial data such as database synchronization for fault and disaster tolerance is not making use of dedicated lines any longer but relying more on the Internet infrastructure for communication.

The Internet infrastructure based on IP was not designed with this task in mind, but rather assumed an unreliable transport layer. TCP (Transmission Control Protocol) has extensive error correction and retransmission capabilities that accommodate this unreliable transport network. References [33], [34], [35] and [36] discuss extensively the issues that need to be addressed in data and optical networks for proper operation of TCP/IP and constitute a most complete set of references and guidelines on those issues at the different hardware layers, both wireless and wire-line, electronic and optical. This work made use of these references extensively to build on a more complete understanding of the needs in both IP and optical networks, therefore being able to draw a more complete picture of the task at hand and the solutions presented. Focused discussions on IP and optical networking can be found in [42] and [43]. Reference [42] describes architectural alternatives for interconnecting IP routers over optical networks, taking into account signaling as well as routing issues. It also describes how IP-based protocols can be used for dynamic provisioning and restoration of lightpaths, and the different issues of relevance to the interoperability between differing optical hardware equipment. Reference [43] focuses on the different switching techniques and technologies in optical networks, presenting both transparent packet networks and Optical Burst Switching (OBS). It identifies the following three points of relevance to optical switching networks:

- Self-similar nature of Internet traffic, described extensively in [44], where Internet traffic exhibits the same characteristics regardless of the number of sessions or the

time sampling granularity.

- Routing and data flow asymmetry, where client Internet downloads are much greater than the data uploads.
- Server-bound congestion, where even though fibers may be available throughout the network, server congestion will deliver slower service. Internet caching companies such as MIT-originated Akamai ®, have been trying to solve this problem by distributing servers across the network.

These problems have to be taken into account when dealing with improvements to the existing network architecture, problems that we will address in this document. Optical fibers have been deployed extensively over that past few years that provide faster transmission speeds and much smaller error rates. This has created a transmission infrastructure that is mostly unused (dark fiber).

In addition, fast Internet service to the home has become a reality. Cable modem technology, DSL as well as broadband wireless have become a reality, gaining a market share of about 4% of all home access in the year 2000. If one try to understand the reasons behind this low penetration rate, several issues arise that are related to pricing, economics and technology. The actual hardware and software implementation of these technologies is not very challenging; these access methods are also consumer-oriented and thus have a high degree of ease of use. The issues on the technology side are that while transmission speed is readily available, switching speed is not. Legacy hardware in the Metropolitan Area Networks running SONET technology and ATM cannot handle the speed that is required in that network, the main reason behind dark fiber.

1.2 Link-State Routing

With the increasing demand for Quality of Service (QoS) over data networks as well as

the convergence of voice and data networks, there is a need to provide service guarantees for networks to allow a high quality of communication. That quality can be thought of in terms of delay, bandwidth guarantee, and dynamic allocation of different levels of service for customers.

In an Autonomous System as defined in [5], a link-state routing protocol called Open Shortest Path First (OSPF) takes care of the routing decisions as well as recovery in the case of link failure. The OSPF implementation does not make intelligent decisions on routing based on information about utilization, link speed or overall route attractiveness. While the simple protocol performs well, it fails to make use of other layers' information to lead to a better network utilization, and ultimately better customer service. Extensions to OSPF to allow better routing and failure recovery have been proposed that rely on making judicious link cost assignments. Link cost allocation allows OSPF to find different shortest paths based on the current network state, network utilization and expected future demand. The motivation for the work is to make unused routes more attractive, distributing load over the network resources. Those methods lead to mixed results, some only trying to solve the problem of static bandwidth allocation, while others relying on measurements to make route allocations. A more thorough discussion of the different techniques as well as their respective advantages and drawbacks is presented in Sections 2.2 and 3.2.1.

Cost changes on links in OSPF often lead to traffic re-routing. Unrestricted re-routing causes routing loops and dropped packets, which then leads the TCP protocol to go into congestion control and congestion avoidance. This leads to transmitting node to decrease the speed at which it sends data through the network, ultimately decreasing network utilization. This works against the original intention of the described extension

methods. While small changes do not affect traffic much, important topology changes may lead to large cost changes. This is especially true in the case of link failures. Part of this dissertation aims at solving these problems and presenting a more complete solution to the challenge of traffic engineering in the Internet.

1.3 Traffic Engineering in Data Networks

Traffic engineering is the ability to make use of network resources intelligently to support data transmission requirements. One of the advantages of traffic engineering is the ability to have connection-oriented traffic, in other words, the ability to establish data paths that meet a specified bandwidth need [31]. This increases the reliability of traffic delivery because knowledge of the network resources allows traffic-engineering algorithms to make intelligent routing and delivery of data.

1.4 Scalability in Switching Systems

Switching systems have historically consisted of a buffer at each input port, a shared memory that allows the switching and buffers at output ports. This design allows efficient switching of data from an input port to an output port. Data in the past few years has moved from being electronically encoded on copper or coaxial cable to being transported in fiber optics through the use of lasers. This has led to the need for expensive optical-electronic and electronic-optical converters. At each input port an optical-electronic converter converts photons to electrons. At each output port, data is converted from electronic to optical.

With the advent of Wavelength Division Multiplexing (WDM), more wavelengths can now be transmitted over the same fiber with minimal crosstalk. Dense Wavelength Division Multiplexing (DWDM) is the ability to pack even more wavelength on one fiber. Lucent has been able to demonstrate up to 1,022 channels on a single optical fiber

[37]. Both channel capacity and density are expected to grow further in the next few years. A demultiplexer and a multiplexer in the switch separate the wavelengths from one another at the input ports and pack them together at the output ports respectively -a graphical depiction is shown in Figure 4-1. While those achievements are remarkable, the limited number of wavelengths that can be made commercially available (40-80 using technology available in year 2001) requires that data from different sources and going to different destinations be packed on the same wavelength. If technological advancement allowed for use of as many wavelengths as there were flows in the network, the switching problem could be solved readily through the use of all-optical cross-connects, switching systems that switch whole wavelengths.

This has left switching systems with the hard task of scaling to accommodate all these channels. Opto-electronic conversion equipment is expensive and bulky, making the task of putting a thousand converters in one switch almost impossible, even with the miniaturization of this equipment. On the other hand, the electronic switching core could not sustain the speed at which data would be switched, nor would the large buffers needed be realizable with today's technology.

To deal with both issues, network equipment manufacturers have adopted one of two different technological choices:

- Miniaturization and parallelism of existing hardware equipment by making components smaller and by racking multiple switches and implementing efficient clustering solutions to accommodate the speed increase.
- Use of all-optical cross-connects, equipment that can switch light without the need for conversion equipment. Optical cross-connects, or OXC's, are based on different technologies and allow the switch to direct a wavelength coming on a

certain input port to leave on a certain output port. Optical wavelength conversion that allows a switch to convert from one wavelength to another has just recently become a reality, but is lossy and expensive.

The first strategy's advantages are the ability to build on proven technology and expand on it to deliver better and faster transport. In addition, the ability to mix traffic data and send that fine granularity data on any wavelength is of great benefit. Disadvantages include the reliance on frequent hardware upgrades, high cost of deployments and important scalability problems.

The second strategy allows for great switching speed (theoretically limited to the number of ports that can be switched). In principle, data coming in, no matter how fast, would be switched seamlessly in the optical cross-connect that would not require frequent upgrades. The drawbacks include the high cost of optical wavelength conversion and the inability to deal with data at the flow granularity but rather at the grain size of a wavelength: since there is no way to inspect and route packets optically, the switch can only make the decision to switch a whole wavelength from a determined input port to a determined output port. This leads to problems in grooming[1] data over already assigned wavelengths and retrieving data out of wavelengths. Both these operations require the termination (optical-electronic conversion) of the wavelength to be able to add/retrieve data. The all-optical switches act as ideal circuit-switched networks.

If we try to rationalize the architectural differences behind data and voice networks, we can see a pattern of two data types that justify on one hand circuit switching such as ATM and SONET switching, that support voice traffic particularly well and on

[1] Grooming is the operation of packing different low-speed traffic streams into multiple, high-speed wavelength channels in a WDM network with the goal of reducing equipment usage

the other hand packet switching, such as IP routers. Traffic patterns have changed over the past years, but the suitability of packet switching and circuit switching to several different traffic types remains the same. Voice is still ideally switched through a circuit while web traffic is best switched at a packet level. Other examples of data that are better switched when the network uses a circuit are large transfers of data such as copying very large files over the network (as in File Transfer Protocol (FTP) traffic) or database synchronizations between different corporate offices.

The choice of one switching method appears to be motivated by the preference of the hardware equipment manufacturer and the choice of the telecommunications carrier based on past experience, stockholder requirements and strategic positioning as a flavor of the day.

This work set about understanding traffic patterns and the implications of those patterns on present and future network infrastructure requirements. This led to a two-pronged approach to understand what would ultimately make an appropriate switching infrastructure.

The thesis addresses the delivery to data networks of QoS assurances and guarantees usually enjoyed by voice networks. We consider several promising emerging technologies including Differentiated Services (Diffserv) and Multi-Protocol Label Switching (MPLS), to understand their value and applicability. This work leads to proposing a methodology for traffic engineering that uses Diffserv and MPLS to provide quantitative QoS guarantees in data networks. It presents algorithms and mechanisms that enable the types of resource reservations that voice networks deliver. The model applies within an Autonomous System (AS) as defined by the OSPF link-state routing protocol [5]. It makes use of a Centralized Resource Manager that knows of the resource

availability and utilization and accepts or rejects calls based on availability.

Based on this study, the first part of the thesis looks at networks where the OSPF link-state routing is already deployed and the implications that QoS routing has in these networks. In networks that make use of traffic engineering by optimizing link costs, changes in those costs may lead to routing loops. A mechanism to prevent that is discussed and its correctness is proved.

The second part deals with the hybrid switch. Based on the experience we built in the other sections, the model of a switch that combines both optical cross-connect and IP router is proposed. Optimizing traffic based on intelligent cost assignments allows the switch to optimally route traffic to its destination. An Integer Programming formulation is proposed that solves the static part of the problem. This Integer Program uses pre-determined demands in a network where resources are based on the nodes, links and bandwidth available to solve the objective function of routing all traffic optimally while respecting physical as well as networking constraints. Physical constraints include the inability to inspect optical packets of data. Networking constraints include the need to keep every flow on one path, without doing load balancing that flow or parts of that flow on the network resources.

A heuristic algorithm is also described that solves the routing problem for dynamic demand coming at the access points of the network by making intelligent use of both switching cores within the hybrid system. This heuristic algorithm makes the appropriate routing and wavelength assignments while distributing load across the resources of the switch and the different links in order to allow for future traffic requests to be accepted. A simulation study is presented to assess the behavior of the heuristic algorithm, areas of interest especially in terms of load balancing and link utilization as

well as ways to enhance its operation.

1.5 Thesis Outline

This thesis describes the work conducted in implementing QoS guarantees over data and voice networks. It leads to the proposal of the hybrid electronic and optical switch to achieve a good distribution between packet and circuit switching. It introduces the issues that arise in this domain, the ways we address them, and the results that we obtain. It also presents future research that could be conducted in the area of switching and routing in general and measurements for optimizing the heuristic algorithms in particular.

Chapter two describes an architecture devised to support Diffserv traffic especially Expedited Forwarding (EF) traffic, by making appropriate resource allocations to deliver the required QoS guarantees, in terms of bandwidth, delay and jitter. It presents earlier work and approaches in the literature and builds on that experience to build a better and more robust architecture for the needs of QoS traffic. The framework proposed achieves that objective by delivering service that approaches the quality of voice networks both in terms of QoS reservations as well as restoration –a mechanism that finds alternative routes in the case of link failures due to either a fiber cut or transmission or equipment breakdown. This architecture is used to provide a breadth of QoS service guarantees over data networks. This increases the returns of traditional Internet Service Providers by using a data network infrastructure as opposed to requiring that they use a voice infrastructure.

In chapter three we describe the QoS extensions to routing in OSPF that deliver transmission of the general routing problem, which is an NP-hard problem. In this case, link failures that may lead to routing loops are taken care of through an algorithm that we devise to deal with cost changes that are the main reason behind these routing loops.

This discussion presents several different techniques of restoring network connectivity due to link failures, and shows the shortcomings of all these techniques. The algorithm that we propose does not have any of the shortcomings and leads to optimal routing. The correctness of the algorithm in the face of single link failures is proven and the optimality issues that may arise are discussed. This allows us to approach the hybrid switch problem with knowledge that OSPF with QoS extensions is survivable and leads to a good distribution of the network load among the available resources. Restoration is possible in this network by using the results of chapter three and the failure mechanisms deployed in QoS-enabled link-state routing.

Chapters four and five are the presentation of the hybrid switch design. We discuss the shortcomings of each of optical switching and electronic switching. We also discuss the advantages of each of these switching techniques. This leads to proposing a hybrid switch. Traffic coming through the network is either based on static pre-arranged Service Level Agreements that can be best served through an Integer Programming Formulation presented in chapter four. Chapter five presents an algorithm in which dynamic traffic requests come at the ingress (access) nodes in the network and are accepted or rejected based on available resources. This discussion clearly shows the advantage of the hybrid method over all-optical networks and the ability of hybrid switches to service traffic more efficiently (database and voice traffic using circuit-switching) and effectively (the algorithm achieves good network load balancing and increased utilization of the wavelengths). Simulation work that indicates the ability to achieve a good network load and load distribution is also presented in chapter five.

Chapter six summarizes the motivation and accomplishments presented in the earlier chapters, while also providing an extensive agenda of open issues that, if

addressed and resolved in future work, would advance the state-of-the art in data communication networks. Optical networks are poised to play a central role in the delivery of next generation services to the users that seeking more bandwidth and faster response times. Of particular interest is the ability to perform dynamic traffic management that would allow a more selective approach to determining what flows would be best serviced through optical switching and what other flows would be best serviced electronically.

Chapter 2

Traffic Engineering Algorithms Using MPLS for Service Differentiation

2.1 Introduction

This chapter proposes a Traffic Engineering methodology that uses Diffserv and MPLS to provide quantitative QoS guarantees over an IP network based on the PASTE architecture proposed in [13]. We provide mechanisms and algorithms that will enable a service provider or network operator to make resource reservations [38]. The model uses a network-wide aware approach in making decisions. A Centralized Resource Manager (CRM) keeps track of an Autonomous System's (AS) resources and accepts connection requests by setting up Label Switched Paths (LSP) that will service that request with the necessary resources.

The chapter presents an algorithm that deals with the changing resource needs of

an already existing LSP. The architecture provides the ability to do traffic restoration due to resource failure by keeping a list of candidate paths at the CRM that can be used in that event. A discussion of the restoration capability of the algorithm and its extensions is also presented to show the applicability of high-quality services such as those available in the phone network (low delays and restoration being some of these services) to data-centric networks.

2.2 The Need for Traffic Engineering

The Internet Engineering Task Force (IETF) is working to produce protocols to support differentiated Quality of Service (QoS) on IP networks. Currently the Internet treats all data in the same manner, making no differentiation based on the source/destination or nature of the data. The motivation behind this equal treatment is to allow the network to deliver best-effort service to any packet that crosses through it. The goal, however, is to develop the infrastructure to better service new IP-based Internet applications that have specific requirements. For example, voice data is intolerant to excessive time delay or jitter. Conversely, the processing of a large financial transaction may be tolerant to moderate delays but have a very large bandwidth demand.

2.2.1 Differentiated Services

Contributors to the IETF envision a next-generation Internet that can offer choices to customers and applications as to the treatment of their data. Towards this goal, the IETF has proposed the Differentiated Service (Diffserv) architecture to enable IP networks to support multiple QoS needs [1].

Interior nodes and boundary nodes are grouped into an Autonomous System (AS). An AS consists of a group of nodes administered by a single entity. A Diffserv domain is defined in [1] as a "contiguous set of DS nodes that operate with a common service

provisioning policy and set of PHB (Per Hop Behavior) groups implemented on each node". For the purposes of this discussion, we have assumed a single DS domain within each AS. This simplicity allows us to deal with the delivery of *diffserv* in networks without the need to deal with implementation details for inter-domain communication. Source/destination pairs may directly connect to a single Autonomous System as shown in Figure 2-1, or traverse more than one AS.

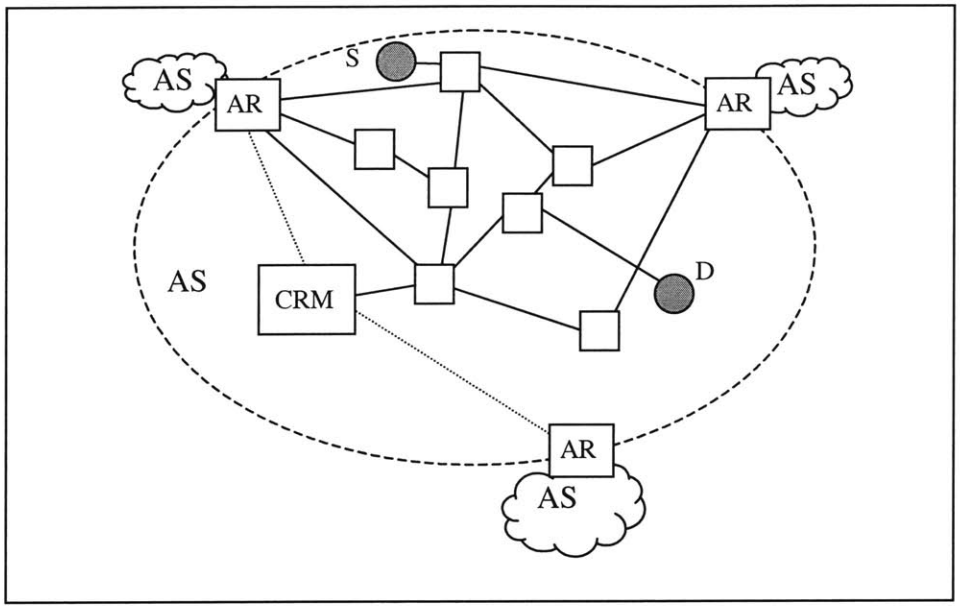


Figure 2-1 Autonomous System of an Internet Service Provider and its relationship to other networks

The IETF has defined one PHB and a PHB group, namely the Expedited Forwarding (EF) PHB [3] and the Assured Forwarding (AF) PHB Group [2]. The PHB is assigned a certain Behavior, defined as Behavior Aggregate, which defines its treatment in the network. Examples of end-to-end service using the EF PHB include Virtual Leased Lines (VLL) [3]. The "Assured Forwarding (AF) PHB group is a means for a provider's DS domain to offer different levels of forwarding assurances for IP packets received from a customer DS domain." [2] Reference [1] outlines two other architectural building

blocks, Traffic Classifiers and Conditioners, and Network Resource Allocators. Traffic Classifiers and Conditioners protect interior nodes from resource starvation. Generally, the DS domain services flows under a Traffic Conditioning Specification (TCS) decided between the domain and its flow's sources. Violating flows that arrive at an ingress node will be dropped, shaped, or remarked as defined by the TCS. Dropping traffic is deleting packets of that traffic without forwarding them further. Shaping is the act of making traffic arrival and departure conform to a certain rate and distribution.

2.2.2 Diffserv and MPLS

An internal Diffserv node treats all packets of a particular Behavior Aggregate identically. If a particular customer's flow shares the same DSCP with other flows, it is difficult to characterize the treatment of a customer's packets at an output port without knowing the number of other flows with the same DSCP. A customer's perceived end-to-end service will be a function of the service received at each node along its path, and thus is even harder to characterize.

One protocol which is capable of specifying, or "pinning", a flow's route that provides quantitative guarantees is Multi-protocol Label Switching (MPLS). MPLS is a protocol that can create tunnels between a pair of nodes. An IP packet traversing an LSP is prefixed with an MPLS header. When a router receives a packet with an MPLS header, it uses a separate MPLS forwarding table to determine the next hop. This closely emulates the operation of a circuit.

In summary, MPLS will pin a particular route for a flow determined by a Network Resource Allocation process. MPLS will specify a next hop and diffserv will specify the treatment of a packet waiting to make that next hop.

2.2.3 The Resource Manager

A Centralized Resource Manager (CRM) is proposed to provide Network Resource Allocation. It becomes the primary contact when a customer wishes to initiate a new or expanded TCS. While the characteristics of a flow might change, the CRM acts only when a customer wishes to change its TCS. As an example, a customer may request the DS domain to support traffic between nodes *A* and *B* that will support 30 IP Telephony conversations. The CRM would be responsible for finding a path between A and B. While the flow's characteristics may change over time as calls are instantiated and torn down, the TCS would not change.

The CRM knows the network topology from the system administrator or from the link state descriptors advertised by each node running Open Shortest Path First (OSPF) [5]. The CRM also maintains a database containing the unreserved resources at each output port of each node available for flows with quantitative QoS requirements.

As it creates a path for a flow with quantitative QoS requirements, the CRM follows the following steps:

1. When the CRM receives a request for TCS with a QoS requirement, it determines a set of possible routes, and picks a route that meets the QoS requirement.
2. Once a path has been identified, the CRM must assure that the flow follows this path. Appropriate MPLS label-switched label distribution should be used.
3. The CRM updates its database of available resources to reflect the allocation for the new flow.
4. The CRM signals the ingress router with the information needed to mark and police the new flow and informs the customer that it can send data into the network.
5. The CRM will continue to review OSPF link state advertisements to detect any link

failure.

At initialization, the CRM will have knowledge of the resources available for quantitative TCS's. This database does not hold all resources available at each node, but only the resources reserved by the network operator for flows requiring quantitative guarantees.

2.3 QoS Routing: Building Feasible Paths

This section describes the extensions to link state routing protocols such as OSPF (Open Shortest Path First) and IS-IS (Intermediate System-Intermediate System) to support QoS routing.

2.3.1 Extensions to Support QoS in Link-State Routing Protocols

Interior Gateway Protocols (IGP's) are responsible for routing and route update. They route data by selecting the path with the least cost. OSPF is one such IGP. The most frequent implementation of OSPF allocates unit cost to all links, leading the cost function to pick the least number of hops as the shortest path. Problems arise when:

1. Multiple streams converge on specific links or nodes
2. A traffic stream is routed through a link or node that lacks enough bandwidth to service it [6].

Extensions such as those proposed in [7] and [8] to support QoS routing based on OSPF have been proposed to take into consideration both aspects of the problem. QOSPF [8] is a proposed extension to OSPF to support QoS by flooding the network with information about the available and used link resources. The proposal makes routing decisions based on topology, link resources available and traffic requirements. The QOSPF framework uses the ReSerVation Protocol) (RSVP) [9] for signaling, allowing ingress routers to send the QoS requirements for incoming traffic in an RSVP PATH message. If a QoS route

can be computed and a path reserved, an RSVP RESV (RESrVation) message is sent back, reserving the resource and accepting the request. Another approach that we call QoS-OSPF [7] uses measurements to keep individual nodes' view of the network updated. QOSPF [8] bases its calculation on state information rather than measurement. RSVP is used to communicate QoS requirements to each node. Both QOSPF and QoS-OSPF choose a route by solving a shortest path algorithm using link costs dependent on available resources. Consequently, the time between runs of the Dijkstra shortest path algorithm is much smaller than in OSPF, creating a higher computational burden.

In the case of QoS-OSPF, the nodes determine their available resources by direct measurement, and then flood the network with this information. Since the amount of available resources changes rapidly, especially in the context of bursty Internet traffic, QoS-OSPF generates a substantial communication overhead. QoS-OSPF tries to minimize this overhead by using a trigger mechanism. Triggers at a node fire every period T, or when a link resource has changed by a given percentage. Another potential problem occurs when QoS-OSPF measures underutilized but allocated resources. These resources could be reallocated and cause packet drops once the client starts using the full Virtual Leased Line (VLL) allocation.

Our approach addresses the above stated problems by making a CRM responsible for all resource allocations. The CRM relies on its view of the AS, the available resources and the reservations it has accepted to service QoS requests. Many issues of signaling overhead and delay are avoided.

2.3.2 Algorithms: Paths and Alternate Paths Pre-Computation

The CRM first determines the shortest paths between all ingress points, by running Dijkstra algorithms starting at each node. When they have finished running, the CRM

knows the shortest path between any two nodes. The CRM then runs a series of Dijkstra algorithms with a modified topology. The number of algorithm runs for each node corresponds to the number of outgoing links from that node. The algorithm effectively tries to find other candidate paths that do not go through the first link of the shortest path for all source-destination pairs. In order to do this, the CRM sets the outgoing link cost to infinity and runs a modified Dijkstra algorithm. This allows the CRM to find alternative paths starting at nodes that are on the shortest path.

Some nodes will not have second candidate paths besides the primary shortest path ***p*** depending on the connectivity of the graph; such is the case of some border routers that might only have one outgoing link. However, one or more nodes further along that path –such as interior nodes- would find alternate routes for part of the path if they existed. The Dijkstra algorithm will only be run to recalculate shortest paths for the nodes use the link that we consider to be down. Although the worst-case order running time remains the same, in practice the Dijkstra algorithm ends faster. The Connection Admission Control (CAC) decision is explained in Section 2.4.

2.4 Connection Admission Control

In this section, we describe the process that the Resource Manager goes through to either admit or reject a connection request.

2.4.1 Meeting Traffic Requirements

The problem of finding a path that satisfies several QoS constraints is NP-complete, but polynomial-time algorithms can be used if one assumes that the network service

disciplines are rate-proportional[2] [10]. An example of such a queuing discipline is Weighted Fair Queuing (WFQ) [11] also known as Generalized Processor Sharing (GPS).

Assume the aggregate traffic source is constrained by its leaky bucket parameters (σ, ρ) where σ is the maximum burst size and ρ is the average token rate. Assume a path $\boldsymbol{p}$ of n hops and link capacities C_i at hop i. Let the residual bandwidth on any link i be R_i. Let L_{max} be the maximal packet size in the network, $prop_i$ the propagation delay at hop i and r the amount of bandwidth requested ($R_i \geq r \geq \rho$) for all $i \in \boldsymbol{p}$. The following bounds based on work in [11] have been found to apply.

The maximum end-to-end delay bound is given by:

$$D(p,r,\sigma) = \frac{\sigma + n \cdot L_{\max}}{r} + \sum_{i=1}^{n} \left(\frac{L_{\max}}{C_i} + prop_i \right) \tag{2.1}$$

Delay jitter is bounded by:

$$J(p,r,\sigma) = \frac{n + L_{\max}}{r} \tag{2.2}$$

Buffer space requirements at hop i are bounded by:

$$B(p,\sigma,i) = \sigma + i \cdot L_{\max} \tag{2.3}$$

2.4.2 Selecting the Path

Given those upper bounds, the algorithm needs to verify one or more of the following conditions to meet the respective bounds.

[2] Rate-Proportional Servers (RPS's) constitute a type of scheduling algorithms that can provide upper bounds on end-to-end and delay guarantees when the burstiness of the traffic is bounded. Session traffic can be done bounded by the use of shaping through a leaky bucket.

$$
\begin{aligned}
&D(p,r,\sigma) \le D_{requested}, \\
&J(p,r,\sigma) \le J_{requested}, \\
&B(p,\sigma,i) \le B(i)_{available}
\end{aligned}
\tag{2.4}
$$

On the other hand, there are instances where traffic needs to meet a certain delay requirement irrespective of the rate. The maximum bandwidth available on a path $\boldsymbol{p}$ is the minimum capacity of all links l of p. In this case, given the maximum delay requested $D_{requested}$, and $c_{\max}$ on path p, traffic rate and delay should meet the following conditions.

$$
\begin{aligned}
&\rho \le r \le c_{\max} \\
&D(p,r,\sigma) \le D_{requested}
\end{aligned}
\tag{2.5}
$$

Assume a request for setting up a path between ingress router *ingress1* and egress router *egress1*. That request includes the bandwidth r needed, as well as one or more other constraints in terms of delay, delay jitter and buffer space requirement. The CRM maintains a structure $\boldsymbol{P}$ of the paths it considers for routing the traffic requested. The CRM first looks at the shortest path from *ingress1* to *egress1*. If any link l has a capacity $C_l < r$, then it is discarded.

If no link has been pruned, the CRM proceeds to the other constraints to check that they do satisfy the inequalities in [1]. If a feasible path is found, the CRM sends a success response to *ingress1* with the chosen path, updates the node resources in its resource database. If a hop has been pruned or if the path does not meet traffic constraints, the CRM adds the available alternate paths between (*ingress1-egress1*) iteratively to $\boldsymbol{P}$ and performs the same checks. If at this point a feasible path has still not been identified, the CRM may explore other paths based on the new paths added to $\boldsymbol{P}$.

For each of the paths in $\boldsymbol{P}$, the CRM iterates over the hops i and selects alternate

routes between (*ingress1*; *hop i*) on one hand then (hop i; *egress1*), while performing the CAC. For a path ***p*** identified by its hops: ingress1-node1-node2-node3-egress1, the algorithm tries to perform the CAC on the following paths, while adding them to ***P***.

- `ingress1-alternate route(s)-node1-node2-node3-egress1`
- `ingress1-node1-alternate route(s)-egress1`
- `ingress1-alternate route(s)-node2-node3-egress1`
- `ingress1-node1-node2-alternate route(s)-egress1`
- `ingress1-alternate route(s)-node3-egress1`
- `ingress1-node1-node2-node3-alternate route(s)-egress1`

The intuition behind this technique is that since the CRM rejects the path, it is because of either a lack of capacity or one of the other QoS constraints. A contributing factor is the lack of capacity at a certain link. By using this technique, the CRM is assured that while considering the alternate routes, it circumvents the overused link. Exploring the different possible routes may become time-consuming. A CRM may decide to stop investigating alternate routes after N different routes have been considered. It would try to make multi-trunk selections instead as explained in Section 2.4.4. The algorithm stops at the first acceptable path.

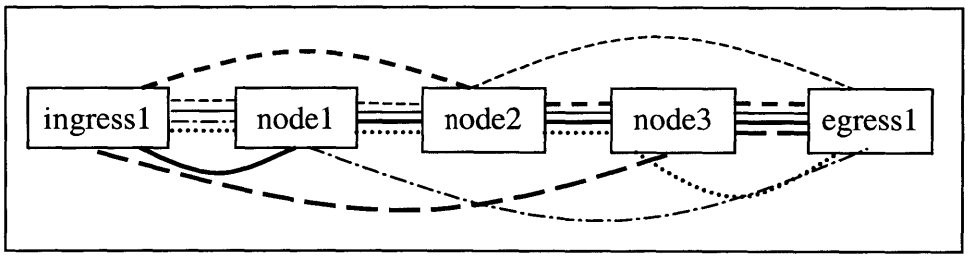


Figure 2-2 Alternate Routes Considered by the CRM

The CRM does not stop when it has checked the alternate routes; rather it forms

possible alternate paths based on the contents of ***P***. It is important that the CRM does not consider a path more than once. To prevent this, it always checks the path ***p*** under consideration against the list in ***P***. The CRM also makes sure that the alternate paths do not lead to cycles. Therefore, ***P*** never contains walks[3].

2.4.3 The Diamond Problem

It is possible that the algorithm fails to discover possible QoS routes. This circumstance is due to a topology where multiple links exist between two nodes or multiple short paths. This leads the algorithm to use up the two links instead of all available links. The logic behind not identifying all possible routes is to limit the processing time required for candidate routes, as well as keep the number of candidate routes small when searching through them. In addition, the Internet backbone topology is a sparsely connected mesh. This kind of topology is more amenable to the solution proposed in this chapter, since nodes have few outgoing links. In fact, several existing metropolitan area topologies are actually constituted of dual fiber ring topologies with two incoming and two outgoing links.

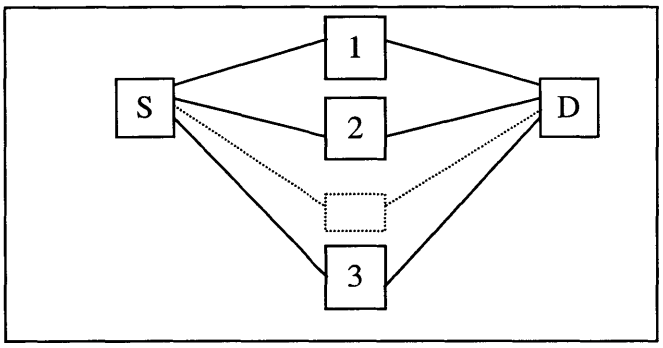


Figure 2-3 Topology Depicting The Diamond Problem

[3] A walk is a set of nodes that are connected by arcs when a node is repeated more than once, thus leading to a routing loop.

Let's consider Figure 2-3. By building the list of paths and alternate paths between any two-node pairs, using unit cost for all links, the algorithm will identify S-1-D and S-2-D as paths between S and D, but fail to identify S-3-D. The CRM will fail to use S-3-D. This problem can be solved. The CRM may identify multiple routes at select nodes, by setting several link costs to infinity and running the modified Dijkstra discussed earlier. Those nodes are the routers with a large number of interconnections such as routers A, B and C on the map in Figure 2-4. As an application to the diamond problem, after the CRM has identified the primary and alternate route, it will set both c_{s1} and c_{s2} to infinity. This identifies S-3-D as an alternate route. Whenever the algorithm finds a suitable path, it should set up an LSP on that route.

If all candidate routes have been exhausted, the CRM sends back a message to the ingress node, notifying it of its failure to pick a route, or tries to make a multi-trunk selection that supports the requested traffic, as discussed in Section 2.4.4 below.

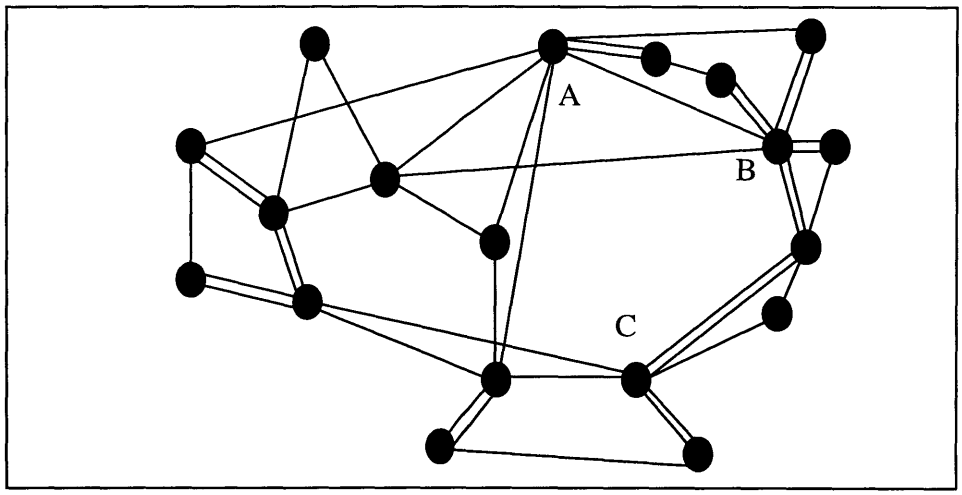
2.4.4 Multi-Trunk Selection

This discussion assumes a network environment where the fine granularity of the micro-flows makes it possible to use multiple paths for routing the same aggregate. It also assumes that the ingress router knows how to route packets on several trunks. The motivation for that is the need to keep packets that belong to the same (source, destination, port) tuple in-order, therefore on the same Label Switched Path. In that manner, packets of the same tuple need not be reordered.

In order to resolve a "false negative" or the unavailability of any one route to sustain the traffic requirement, the CRM may select multiple paths to route the traffic requested. In order to do so, the CRM should consider the candidate paths in $\boldsymbol{P}$ as

explained in Section 2.4.2.

In case we have a bandwidth r requirement, the CRM may solve for the maximum bandwidth available from *ingress1* to *egress1*, by running a maximum flow algorithm. A graph G' constituted of the nodes and arcs in $\boldsymbol{P}$ will be considered when running the maxflow algorithm from *ingress1* to *egress1*. Though implementations of the maxflow algorithm perform at best at $\mathbf{O}$ (n^3) running time [12] (where n is the total number of nodes), the running involves a smaller number of nodes and arcs than the set of nodes and arcs in the whole AS. Let $C_{ingress1\text{-}egress1}$ be the available capacity from running the maxflow



algorithm. If $C_{ingress1\text{-}egress1} < r$, the CRM denies the request; otherwise, it checks whether the other QoS constraints can be met by appropriate distribution of load on select paths considered as follows.

Figure 2-4 MCI Internet Backbone Topology

Equation (2.1) indicates that traffic will experience less delay by increasing its rate used on path p, i.e., when $c_{max}(p)$ is chosen on the path [10]. Therefore, for every path in $\boldsymbol{P}$, the

CRM should try to route the maximum allowable capacity and check against the CAC constraints. Alternatively for a delay-constrained traffic, each path should verify equation (2.5) to carry a part of the traffic. Label Switched Paths are set up that each correspond to a traffic trunk as mentioned in [13]. ***P*** is always reset to an empty set at the end of the algorithm run, irrespective of success or failure.

The approach of pre-computing paths provides a fast solution as opposed to OSPF-based solutions. In addition, communication overhead is reduced using this approach since the CRM keeps track of all resource reservation information.

2.5 Changing Resource Requirements

An issue arises in dealing with an aggregate of flows, as is the case of *diffserv*, since micro-flows should be able to join or decouple from this aggregate dynamically.

2.5.1 Increasing the Requirements of a Traffic Trunk

This discussion deals with the question of providing a trunk with more resources if needed. Such a scenario happens when a corporate network needs to increase the aggregate so that it can accommodate new micro-flows. To do so, the operator (human or automatic agent) sends a request asking for more resources for the aggregate flow. It is the responsibility of the CRM to explore whether it can increase the resources assigned to the traffic.

We propose a scheme whereby the CRM tries to service the extra resource needed on any one of the Label-Switched Paths. Since the algorithm presented earlier may instantiate multiple traffic trunks for a particular traffic requirement, the CRM services the extra flow requirements by adding it to an existing traffic trunk. This saves complexity in terms of running the algorithm of Section 2.4.2. In addition, the CRM does not have to publish a new LSP. The CRM adds the traffic trunks considered to ***P***. If

such increase in resource reservation is not possible when the CAC rejects the flow, the CRM uses the paths in $\boldsymbol{P}$ and executes the algorithm discussed in Section 2.4 to find another route. In case of success, the extra flow will be serviced independently.

2.5.2 Decreasing the Requirements of a Traffic Trunk

By decreasing the traffic requirement, the CRM should not try to move the flow f from the path that it was using to an alternate path that could theoretically support this traffic with the fewer requirements needed. This would lead to out-of-order packet delivery, which is not allowed in the *diffserv* specification [2]. Therefore, the CRM will only update its view of the resources available.

2.6 Restoration: Response to Failure

In a data network, resource (node or link) failure may happen. It is the responsibility of the network to route the data over different routes in order to keep the flow of information undisrupted. The network may find out about a link failure through the OSPF updates. This section describes a method for dealing with link failure using OSPF as the notification agent for link failure. Other routing protocols may be used if the network administrator provisions a mechanism for making the CRM aware that a link has failed. When a link is no longer available, the OSPF update reflects the new network topology, pinpointing the failed link.

At the CRM, this information is crucial for rerouting paths. A CRM receives an OSPF message, updates its view of the topology and knows of link failures. The CRM is responsible for rerouting all LSP's that were using the failed link.

Reference [13] states that different traffic trunks may have different priority. We assume that in the case of a link failure, the CRM selectively reroutes paths starting with the higher priority ones.

Assume that the link (node1-node2) in Figure 2-5 is down. For a path ingress1-node1-node2-node3-egress1, the CRM first considers whether the traffic can use the alternate routes between node1 and node2 by considering the path ingress1-node1-alternate route(s)-node2-node3-egress. Furthermore, the CRM adds the considered path to ***P***. If this is possible, then a new LSP is setup. Figure 2-5 shows the different paths that the CRM investigates.

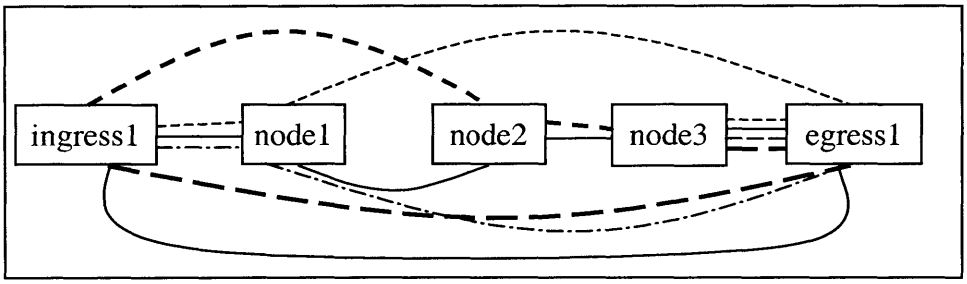


Figure 2-5 Paths Investigated in Response to Failure

- `ingress1-node1-alternate route(s)-node2-node3-egress1`
- `ingress1-alternate route(s)-node2-node3-egress1`
- `ingress1-node1-alternate route(s)-node3-egress1`
- `ingress1-alternate route(s)-node3-egress1`
- `ingress1-node1-alternate routes(s)-egress1`
- `ingress1-alternate route(s)-egress1`

If all these paths fail to sustain the traffic, the CRM would reconsider the paths in ***P***, in a same fashion that Sections 2.4.2 and 2.4.4 suggest. The CRM will examine the other trunks that used link (node1-node2) and perform the same rerouting methodology. If the event a path cannot be rerouted, the CRM should send a connection teardown notification at the ingress node.

In the event where the CRM is successful at moving traffic from one trunk to another, one may consider the packets that have already reached the interior nodes. Since the previous LSP is no longer in effect, interior nodes may forward the packets in the network using IP routing by stripping the packets of the MPLS header. This reduces the number of TCP flows that go in congestion control.

2.7 Recapitulation

For any end-to-end guarantees to be sustained, controlling the flow of traffic through the network is critical. The use of connection admission, intelligent routing, and protection schemes makes end-to-end QoS a much more feasible prospect. Using the functionality provided by Diffserv, and adding to it the route-pinning functionality of MPLS, we can satisfy quantitative QoS guarantees. The proposed Resource Manager offers a solution that removes complexity at the core, without losing control over network traffic. Knowledge of network status allows allocation of network resources, and thus helps in providing better QoS over IP networks.

Chapter 3

Restoration Methods for Traffic Engineered Networks with Loop-Free Routing Guarantee

3.1 Introduction

In a heavily congested network, or because of hardware malfunction as explained in Chapter 2, links can fail or become over-utilized. This often leads to severe rerouting problems, since different routers will build a different view of the network, making the forwarding decisions inconsistent. Paths may therefore include loops, a situation that leads to more congestion, lost packets, and as a result, bad network performance.

Work has been conducted in this field to perform local restoration by letting only a number of routers know of the topology change in the case of a failure. This minimum number of routers, when notified of the failure, will be responsible for routing around the failure, while guaranteeing loop-free paths. This work is very appropriate in the event of

a failure. It fails though to anticipate problems in building alternate routes around congested links. This is due to the fact that it does not keep state of the different link utilizations. Instead, it relies on constant link weights assignments in link-state routing protocols to update routes and restore paths when links fail.

On the other hand, several papers have discussed ways to do traffic engineering, where the routing decision relies on an intelligent decision on the part of the router to avoid congested links and distribute loads around the network, leading to a higher network utilization and a lower probability of congestion. Those works fall into two categories: traffic engineering using new routing and route pinning technologies, such as the MPLS [6] framework discussed in Chapter 2, and traffic engineering based on a more intelligent computation of the OSPF arc weights, based on current and anticipated utilization. Some *ad hoc* techniques set the weight as inversely proportional to the percentage utilization of the different links, but more recent work [15] shows the inefficiency of these techniques. Others have been described in Chapter 2 in the discussion of QOSPF and QoS-OSPF.

Reference [15] calculates a weight matrix based on a supply and demand matrix. This work, while appropriate for long-term requirements, does not deal efficiently with potentially severe changes in topology due to link failures, since it builds on the way OSPF works. The new weights based on the changed topology, known to only the routers that have detected or received notice of the link failure, do not propagate fast enough to guarantee loop-free routing. Those topology changes actually worsen the situation, when based on changing weights, since they occur more frequently.

This chapter discusses a method that combines both techniques, one in building traffic engineered routes, and the other in performing local restoration, to address the

instantaneous need for a change in the topology due to congestion or link failure. The resulting algorithm [39] guarantees loop-free routes that can ensure high network utilization without the need for more complex technologies such as MPLS.

3.2 Previous Contributions and Limitations

This section describes previous contributions and approaches to traffic engineering using extensions to Link-State Routing (LSR) protocols. It also describes the hard problems that need to be addressed by these LSR protocols.

3.2.1 QoS Extensions to the Open Shortest Path First Protocol

Interior Gateway Protocols (IGP's) are responsible for routing and route update. They route data based on the total cost of a path ***p***, which is the sum of costs of the individual links of that path. Depending on the cost of individual links, one can find a shortest path using an $O(n^2)$ algorithm such as Dijkstra's algorithm [12].

Open Shortest Path First (OSPF) is an example of such an IGP. OSPF implementations frequently allocate equal costs of one to all links, therefore picking the path with the least hops as the lowest-cost path. The advantage of OSPF over MPLS is its ubiquitous use and deployment. It is a time-tested protocol that has performed well, albeit not flawlessly. Problems arise when multiple traffic streams converge on specific links or nodes, or when a traffic stream is routed through a link that lacks enough bandwidth to service it without dropping a large number of the packets that it carries [16].

As mentioned previously, both QOSPF and QoS-OSPF suffer from their dependence on measurements and ill allocation of links costs for proper traffic engineering. On the other hand, the use of MPLS as described in Chapter 2 introduces a new protocol that would be a long process to implement, and that would not be able to make use of the currently deployed hardware. In the discussion of the MPLS framework,

a drawback was that the link failure update notifications were still done by OSPF Link State Advertisements.

Recently, Fortz and Thorup [15] have proposed new algorithms to calculate OSPF dynamically, based on a demand matrix, which approximates the efficiency of the general routing problem. The authors publish results based on both synthetic and real networks (the AT&T backbone), with very good results reported on the real network, approaching the performance of the Generalized Routing Problem (GRP). The approach using synthetic networks performs less efficiently. Their work shows that QoS methods using OSPF could approach the efficiency and high utilization that an MPLS-based scheme achieves, without the need to run an NP-hard optimization problem.

3.2.2 Failure Scenarios

The works described in Section 3.2.1 present several methods for setting arc weights that allow the routing protocol to make a smart forwarding decision. Problems arise when the weights change, based on a changing demand matrix, or when an arc fails or becomes severely congested. In this case, some of the nodes notified of the failure/congestion will update their shortest path trees, since the new topology based on the demand matrix will lead weight-setting algorithm to calculate new weights. If all nodes were notified of the arc failure concurrently, they would all pick the same weights. This is not the case in a real network, and the slow convergence leads to the routing loops. Our work deals with the avoidance of routing loops in the case of a routing failure in such traffic-engineered networks.

3.2.2.1 Failure in a network with Constant Arc Weights

Narvaez *et al* discuss in [17] an elegant local restoration algorithm that uses a vector-metric data structure to make forwarding decisions without looping. All routers should

have consistent topology information about the network to avoid loops. Many of the Internet's problems with routing instability are associated with the long delays required to propagate routing information [17], [18] and [19]. The scheme devised by Narvaez *et al* works both for failed links and congested links since a congested link can be represented as if it had failed entirely. The Vector-Metric algorithm assigns to each link a metric represented by a vector rather than a scalar. An element of the vector is defined as the i^{th}-metric. If a failure happens, then the router that knows of the failure will notify nodes on a certain restoration path. The link metrics on the restoration path will be downgraded by moving the array elements a step right. The shortest path algorithm re-computes all shortest paths. The difference between the OSPF restoration mechanism and the Vector Metric Algorithm restoration method consists in the assumption that vector elements with a higher index are infinitely smaller than smaller index vector elements. An issue that arises in such a scheme is that this cost change makes routers see smaller distances to the destination on the restoration path, and therefore would forward data on that restoration path. This may create temporary congestion while other nodes are notified of the link failure. For a complete discussion and proof of the Vector Metric algorithm, the reader should refer to [17].

3.2.2.2 Tunneling

Another approach yet is IP-tunneling, where router A knows of a link AB failure and an alternate path to the destination, so it notifies all nodes of that restoration path. A packet that arrives at A will be tunneled to B through those informed routers then sent normally towards its destination node from B. This leads to bottlenecks since all packets that used link AB now have to go through a restoration path that may not have enough bandwidth. In addition, there is no methodology that can be used for nodes to propagate

link failure information to allow for a better distribution of the network load.

Tunneling is highly processor-intensive; it requires routers to create extra IP headers and calculate their checksum, then process those packets at the end of the tunnel, remove the header, calculate that the checksum is verified, interpret the payload, which is the original IP packet and then forward that packet based on the OSPF forwarding table. IP tunneling does not prevent transient loops from forming when some routers are not informed of the link failure [17].

3.2.3 Problems with Link Failure in QoS Routing

If QoS routing is used in the network, then when an arc failure happens, there is a significant change in the arc weights. Notified routers that use new weight metrics will construct shortest path trees that are different from those of uninformed routers. This may lead to routing loops, as mentioned previously. The packet drops (due to the expiration of the Time-To-Live field of the IP header) or long delays resulting from routing loops lead to a significant amount of TCP traffic going in congestion control and avoidance modes. This leads to a significant underutilization of the network.

On the other hand, a link failure is a severe case that cannot be quantified in terms of expected downtime, and the network should adapt gracefully to the new topology. This requires a long-term solution, while the failure is diagnosed and corrected. The Shortest Path Tree (SPT) before the failure does not apply to the new network topology. Thus there is need to notify progressively and selectively the different routers while making sure that the notified routers can work in tandem with the uninformed routers to produce loop-free routes, with no interruption to the network service.

3.3 Algorithm For Loop Free Routing

Let us first present a few definitions that will allow for a better description of the

algorithm and its operation.

3.3.1 Definitions

We use *link* and *arc* interchangeably in this document, as well as *node* and *router*. A *walk* between nodes A and B is defined as a set of links that connect A through one or several nodes to B, where one node is used more than once. A *path* is a set of links that connect several nodes starting at A and ending at B that does not include any node more than once. An *informed node* is a node that has been notified of the link failure. An *uninformed node* is a node that does not know of the link failure. A node is a *neighbor* of another node when it is linked to that node with an arc. An *informed link* is defined as a link that connects two informed nodes. An *uninformed link* is any other link. A *restoration path* is the initial path that is picked by the restoration algorithm around the link failure and consists of the initially notified nodes. A *restoration network G* is defined to be a set of all informed nodes and links. A *boundary restoration node* or simply a *boundary* node is defined as an informed node that shares at least one arc with an uninformed node. An *interior restoration node* is defined as an informed node that shares no link with any uninformed node. A packet *crosses a boundary restoration node* when it is routed from an informed to an uninformed link or vice versa. We define *Algorithm W* to be the weight-setting algorithm that calculates weights used for optimal routing such as any of the algorithms discussed earlier, running at all nodes.

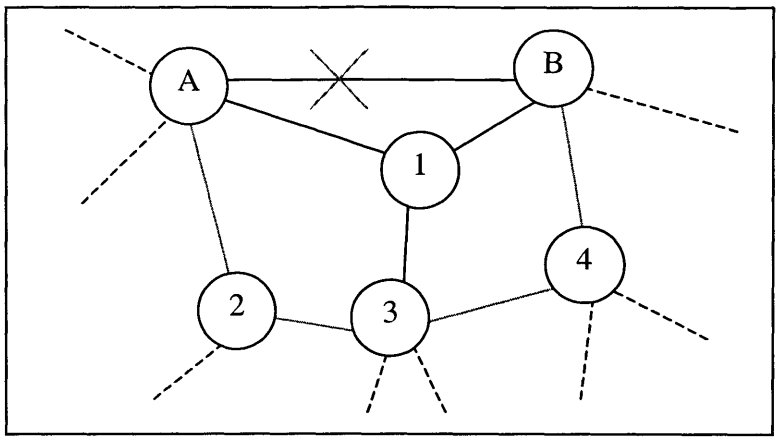


Figure 3-1 Link Failure at AB and Restoration Path A-2-3-4-B

3.3.2 Algorithm Presentation

Assume a network represented by the graph G = (N, A), made of the set of nodes N and the set of arcs A. G uses a set of weights as calculated by Algorithm W. When a link AB fails, a node (one of two nodes A or B that were connected through this link) immediately knows of the topology change and therefore becomes informed. At this time, the routing is sub-optimal since all the other nodes are using the old weight assignments in making their routing decision. The informed node is able to calculate the new set of weights using an algorithm W such as the algorithm described in [15].

This allows the node to build a new shortest path tree. Based on those new weights and with respect to arc AB, a restoration path between AB is picked as the node's new shortest path to the other node. The selection of the restoration path is not need restricted to the shortest path between A and B. Figure 3-1 shows the restoration path A-2-3-4-B used to restore traffic between nodes A and B. All nodes along the restoration path are notified in the same manner as the local restoration algorithm

described in [17], where a special packet travels from A to B, with information about the failure and the restoration path. This approach allows the network to route around the failure fairly quickly. Arcs depicted in gray are the informed links and are set to the new weights, whereas arcs in black are uninformed and are set with the old weights. In this particular example, all notified routers are boundary routers. Any path may be used for the restoration as mentioned previously; running an algorithm for computing the K-shortest paths can be used to choose such a path. It is important though to choose a short path since traffic between A and B that was using the failed link must initially use that chosen path.

3.3.2.1 Forwarding Decisions

Each router makes the forwarding decision independently. The major concern of such an approach lies in the fact that routers have different information about shortest paths, where an informed router may send a packet to an uninformed router based on its view of the shortest path, which may forward it back in a loop. Figure 3-2 shows such a failing case. Weights in black are the old weights before the link failure. Weights in gray are the new weights. B and A (nodes in gray) know of the link failure, and use the new weights to route packets. Node A forwards a packet with destination D to node 1, that forwards it to B based on the old weight assignment. B sees the shortest path to D going through node 1, leading to a routing loop.

The distinction between boundary nodes and interior restoration nodes is the information that a boundary restoration node collects and uses. A boundary restoration node is aware of all boundary nodes. The boundary node keeps its old and new SPT trees. In addition, it knows the old SPT trees of all other boundary restoration nodes, as well as the shortest path trees of its neighbors. To decrease the number of algorithm runs,

the boundary router runs twice an All-Pairs Shortest Paths [12] algorithm that has O (n^3) running time, for old and new weights respectively. When a boundary node becomes an interior restoration node, it can drop its list of boundary nodes and the earlier computed SPT trees rooted at it and other boundary nodes. Basically, an interior restoration node behaves as any regular router.

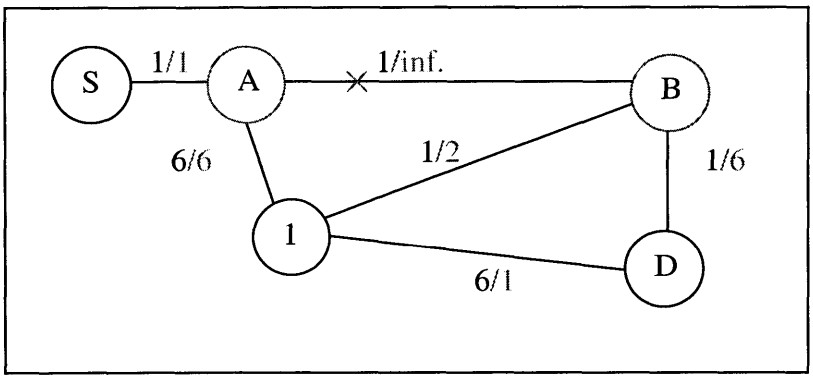


Figure 3-2 Loop Occuring From Different Weight Assignements

It is important to note that the forwarding decisions are not taken in real-time, but rather computed for all destinations based on the different cases discussed previously. This allows routers to build and update forwarding tables that are consistent with their snapshot view of the state of the network. Forwarding tables of boundary routers are updated every time a new router is informed of a failure.

1. Packets Coming on an Uninformed link

If a boundary restoration router **R** receives a packet on an uninformed link, this packet may be either crossing the boundary, as defined in Section 3.3.1 or going to another uninformed link. Figure 3-3 shows scenario A with router **R** as the boundary router and a packet coming on the uninformed link (1-R). The router first looks at the old SPT tree. If the shortest path using the old tree has an uninformed link linking it to the

next hop, then there is no boundary crossing. Otherwise, there is boundary crossing.

Case 1: If the packet is going to another uninformed node, the router can forward it using its old SPT tree. Since the packet is forwarded from an uninformed node using the old SPT tree, and the forward path uses the old SPT tree, then this will not result in routing loops in that case. Proof: Since OSPF in steady-state does not introduce any routing loop, a packet traveling using the old SPT tree that does not cross a boundary always hops to a node closer to the destination; forwarding along old paths with shorter distance to the destination creates no loop.

Case 2: If there is a possibility for a boundary crossing, the node should decide what the next hop is to destination *t* based on the new SPT tree. If the next hop *j* based on the new tree is uninformed, it is possible that the old path lead to loops. Router **R** checks the following condition: If D_{old} (j-t) < D_{old} (r-t), then no routing loop will be introduced [17], forward to *j*. Proof: If the distance from *j* to *t* using old costs is smaller then distance from *r* to *t*, this is the only condition necessary to make sure that node *j* will not forward the packet back to **R** and create a routing loop.

Case 3: If for all uninformed neighbors j, D_{old} (j-t) < D_{old} (r-t), it is possible to introduce a routing loop if the packet goes to node *j*. In this case, the boundary restoration node will select one of its informed neighbors i, where D_{new} (i-t) < D_{new} (r-t) that verifies min (d_{new} (r-i)+ D_{new} (i-t)). This allows optimal routing with respect to the new weights inside the restoration network. Two cases should be considered.

Case 3a: If the packet is within the restoration network and there is no boundary crossing, the packet will reach the destination without loops. Proof: Since all informed routers are using the new weight settings, and since OSPF does not introduce routing loops in steady state, routing occurs on a shortest path if informed links are used.

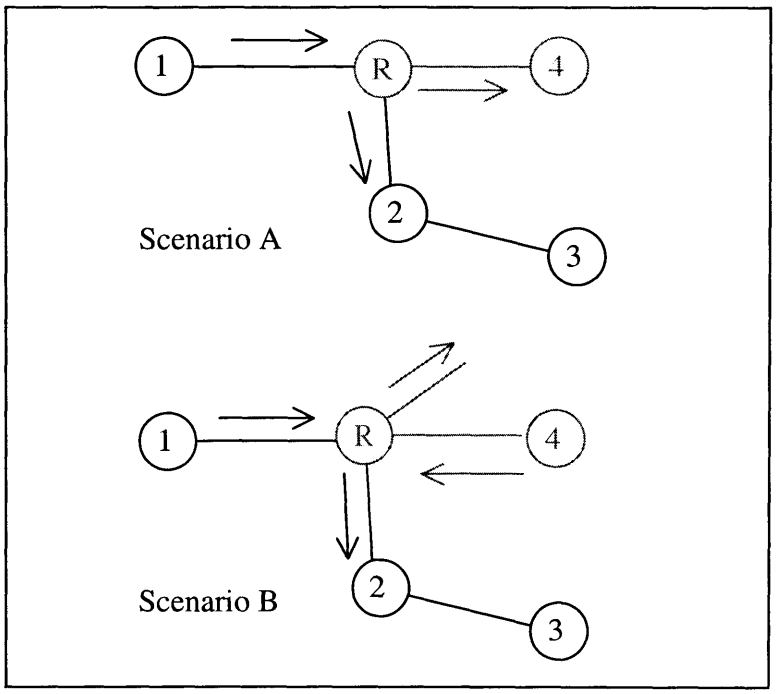


Figure 3-3 Scenarios Related to Crossing Boundaries of the Restoration Network

Case 3b: Router **R** may also send the packet to an informed router if the next hop *j* based on the new weights is informed, even when the destination is uninformed. In deciding whether to forward the packet to an informed router or not, to ensure that no loop occurs, the boundary router will form the path since it knows the routing decision that every boundary router will make. The packet may cross the boundary again one or more times, going from informed to uninformed nodes. The boundary restoration router will examine if this results in a walk and consider two sub-cases.

Case 3b1: If this does not result in a walk, then it will forward it to the next hop, since no loops will occur. Proof: Since traffic is routed between the set of boundary routers based on either old or new weights, and the initial boundary router takes the routing decision based on both old and new weights at every boundary crossing, the

forwarding decision of routers downstream will be the same as router **R**.

Case 3b2: If there is a loop, then it uses its new forwarding table. Since the first boundary router that will receive the packet will make an intelligent decision on either crossing the boundary or dropping the packet, no loop will occur.

Case 4: The last case happens when no informed neighbors satisfy the distance conditions in case 3. From the set of uninformed neighbors *n*, using old distances, one may choose the path with the least distance to destination *t* through the node *i* which verifies **min** ($\mathbf{d_{old}}$ **(r-i)+** $\mathbf{D_{old}}$ **(i-t))**. No routing loops occur. Proof: Since the distance of the next hop to destination is smaller than that of r, then the packet will not return to r, therefore router **R** has not introduced a loop. If none exist, then no path can be used that can guarantee loop-free routing. If the boundary router sees a loop by forming the path a packet will take, it will defer the dropping decision to the exit boundary router. This allows more time for a new informed router to become part of the restoration network, thereby making a loop-free route possible. This behavior is indicative of the fact that a boundary router makes decisions based on its snapshot view of the graph. After the packet has traveled several hops, the original exit boundary router may have changed. Therefore it is beneficial to accept the packet inside the restoration network. Since boundary routers never let a packet leave the restoration network should a loop develop, that forwarding decision does not introduce a loop.

2. Packet Coming on Informed Link to Boundary Router

The discussion in this section closely parallels that of the previous section. If a packet is coming on an informed link, then the boundary restoration router performs several operations to determine whether to send it to an informed or uninformed router. Figure 3-3's scenario B has a packet on the informed link (4-R) coming to boundary

router **R**.

Case 1: Router **R** first looks at the next hop on the shortest path based on the new SPT tree. If that hop lies on an informed router, then it performs the same kind of operation as discussed in earlier where it makes sure that the path the packet follows crossing boundaries 0 or more times will not lead to a routing loop. This approach is due to the fact that even though the boundary router is receiving the packet on the informed link, a boundary restoration node further upstream may have forwarded the packet when several nodes were uninformed, which have become informed since. Proof: Since the router looks at the path downstream to the destination, making the same informed/uninformed decisions the routers will make, it accurately reflects the state of the network at that time and will not introduce loops.

Case 2: If the next hop on the shortest path based on the new weights is an uninformed node, the packet will be crossing boundaries. The boundary node also builds in this case the complete path used to reach the destination. If no routing loops occur, then it will forward it to an uninformed router. Proof: This proof is similar to the proof used in case 1.

If, using the old weights, the next best hop is an informed router, then two cases arise.

Case 3a: If for any uninformed router i, D_{old} (i-t) < D_{old} (r-t), then the boundary router will forward it out of the restoration network to the node i that verifies the following: min (d_{old} (r-i)+ D_{old} (i-t)). This occurs after the boundary router ensures that using that uninformed link will not result in a routing loop.

Case 3b: Otherwise, check that for any informed router *j*, if there exists D_{new} (i-t) < D_{new} (r-t), then choose router j where min (d_{new} (r-i)+ D_{new} (i-t)) and that verifies reach as

explained in cases 1 and 2.

If none of these cases is verified, then dropping the packet will enforce the loop-free guarantee. Since more routers are informed with time, this dropping is only a temporary measure that will assure that no loops develop.

3. Packets Arriving at Non-Boundary Nodes

In this case, the regular routing mechanism should be used, based on either the old (new) shortest paths in the case of uninformed (interior) routers. Their forwarding function does not introduce a loop. Proof: For an uninformed router, the old steady-state forwarding table does not introduce loops. If an interior restoration router receives a packet, then its source was either from within the restoration network and the first boundary restoration node will route it optimally (unless it is the destination); if the packet has originated from outside the boundary restoration network, then a boundary node has made a loop-free decision to send the packet inside the restoration network.

3.3.2.2 Growing the Restoration Network

The first set of nodes that belongs to the restoration network is that picked for the restoration path. To achieve optimal routing after the failure, all nodes are eventually informed of the link failure. In addition, in order to build correct forwarding rules, there is need to keep continuity within the restoration network.

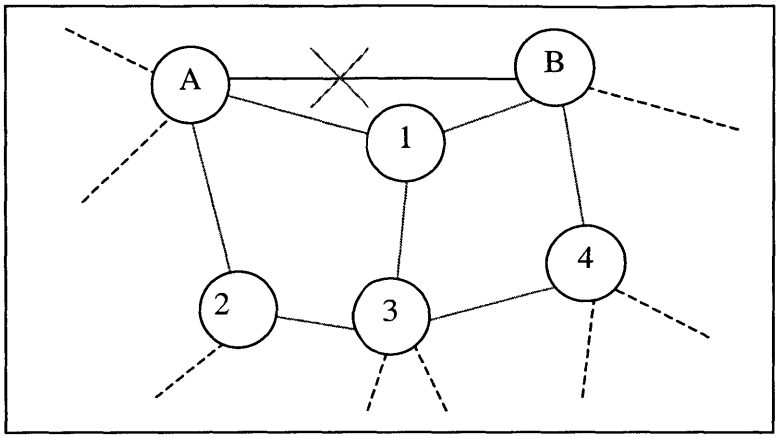


Figure 3-4 Nodes A, B and C notify node 1 of the link failure

These requirements are met by notifying uninformed neighbors of both the link failure and the list of informed routers. These nodes calculate the new arc weights and SPT tree and the SPT trees for the different boundary nodes using the All-Pairs Shortest Paths algorithm. They then turn into informed nodes. No synchronization issues arise from this notification since a packet does not leave the restoration network should a loop develop. A new boundary router updates the other boundary routers of its new state. Every informed node whose neighbors are all informed of the link failure becomes an interior node, therefore, dropping all state information about other boundary nodes and about its neighbors and their SPT trees.

Figure 3-4 shows such a scenario. This scenario closely follows the scenario depicted in Figure 3-1. In this case, nodes A, B and C notify node 1 of the failure of links AB. Since node 1's neighbors are all informed, then by definition, it will become an informed node, and also an interior node. In this case, node 1 only runs a shortest-path algorithm to update its forwarding table based on the new link costs and acts as a regular

routing node.

3.4 Theorem

If when making their forwarding decision, assuming boundary routers are running the same algorithm W and are aware of the other nodes in the restoration network, the aforementioned forwarding rules will ensure no routing loops.

3.4.1 Proof of Correctness

All routers are running the same algorithm W; therefore, all routers will calculate the same arc weights as the calculation is based on the same demand matrix and network topology. The cases of either boundary, interior or uninformed nodes have to be taken into consideration. Since every forwarding decision made by the boundary router does not introduce a routing loop or result in a transient routing loop occurrence, as shown for all the different cases considered, then no routing loop is introduced by the forwarding algorithm presented in Section 3.3 of this thesis. Since interior or uninformed nodes forward data using their shortest path trees, these shortest paths based on OSPF do no introduce routing loops, and therefore the interior or uninformed nodes do not introduce local routing loop either.

3.4.2 Discussion on Optimality and Correctness

Optimality is based on the cost assignments, network topology and load distribution. When a link fails, the optimality of the routing is lost at that point. The approach described aims at achieving optimality again. For that purpose, it starts with a sub-optimal but feasible solution and achieves optimality while staying feasible. It is the intent of this method to achieve the optimality of routing based on the algorithm W. Several path choices may not lead temporarily to a good utilization of the network resources, both inside and outside of the restoration network. This is due to the

conflicting snapshots of the resources. In the restoration network, routing is optimal with respect to the new weights, not the complete topology. This implies that optimality will be achieved when all members of the network are aware of the link failure.

3.5 Recapitulation

We have discussed a method for allowing path restoration in the case of a link failure or congestion. The forwarding rules present in the routing process rely on both the old view of the network and associated weights, as well as the new topology and weights. The forwarding rules are proven to be correct and lead to loop-free routing. Weight assignment is based on a certain network topology, demand matrix, and weight-setting algorithm, ensuring the same weight calculation at each of the nodes. This allows us to tackle the problem of building a better switching architecture that makes use of lessons learned in link-state routing, the new ability of dealing with link failures efficiently and in a loop-free manner and the availability of traffic engineering methods such as the ones described in Chapter 2 to deliver bandwidth guarantees, delay and jitter bounds.

Chapter 4

A Hybrid Optical and Electronic Switch Framework

4.1 Introduction

In today's fiber optic communications networks, switching occurs in the electronic layer. Optical signals are converted into electrical signals, switched to the right channel, and then converted back into optical signals for transmission. This multi-step approach to switching is not only costly but also complicates network design. It is not suitable for the next generation optical network that has to transport hundreds to thousands of times more bandwidth than today's network. Electronic switches cannot accommodate the sheer number of packets that they would have to process in the next generation networks. Figure 4-1 shows the steps required in both an IP router and a DXC (Digital Cross Connect: switches with an electronic core, also called Electronic Cross Connects or EXC). The different wavelengths are de-multiplexed, each wavelength is converted to an electrical signal using an Add Drop Multiplexer (ADM), the signal is switched through

the IP router, the electrical signals are converted back into the different wavelengths using the ADM and multiplexed again. They are together forwarded downstream on a certain fiber.

Electronic switching is inherently constrained and cannot take advantage of the speed allowed by optical fibers, due to the speed limit of electronic equipment. One way to circumvent that constraint is to use parallel electronic switches. This increases the design complexity by requiring complex packet scheduling as well as parallel operation of the assembled switches to achieve the high bandwidths sought. A DXC requires one ADM per wavelength and each ADM can sustain a certain traffic speed. With deployed fibers carrying today 40 channels (wavelengths) and beyond, traffic speed constantly growing (twice as much data carried every 6 months), increasing the number of ADM's per switch and changing those ADM's for new speed limits becomes very expensive.

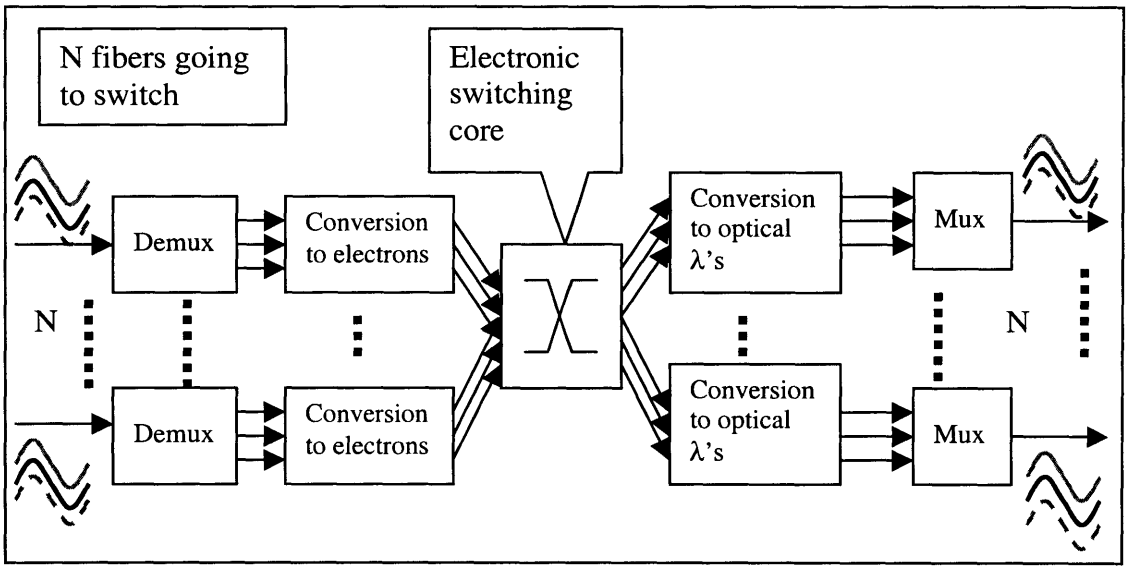


Figure 4-1 Limited and Component-Intensive Switching

Optical switches in theory simplify the network design by requiring less

hardware. They perform the switching function at the wavelength level. They are currently unable to read data at the optical layer though, which prevents them from doing packet switching. Instead, they are limited to wavelength switching. Therefore, an optical switch (or Optical Cross Connect: OXC) allows a coarser granularity but higher speeds than an electronic core (IP router or Electronic or Digital Cross Connect: EXC or DXC) allows. The issue of network topologies is also important, with all-optical networks allowing mesh topologies. This compares to the currently deployed SONET and WDM technology that require a ring topology. Mesh topologies are more flexible geographically but still allow the traffic protection that ring networks do. Our work focuses primarily on mesh topologies, and then discusses the implications for ring topologies. By geographical flexibility, we mean that there are topology constraints due to several factors outside the control of the carrier or Internet Service Provider. These include rights-of-way on bridges, highways, and the existence of bridges over certain rivers, tunnels, mountains, etc. The issue is even harder to deal with in the case of transoceanic cables. In that case, these fiber cables need to be laid next to shipping routes to allow for maintenance and upgrades. Forcing a ring topology becomes inefficient. On the other hand, protection is important at all times, since the carrier needs to make his network fault-proof, therefore requiring multiple paths between nodes on the network. Protection is the mechanism by which several (two in general) paths are reserved to service the bandwidth needs of customers. These paths are node-disjoint, meaning they do not share any node. One of the paths will be the primary path serving voice and data traffic. The other path acts as a failover path in the case of a fiber cut or a node down on the primary path. That path will take over data transmission in that case in a minimal amount of time.

Section 4.2 discusses previous work of relevance to this chapter. We will make present several ideas that facilitate the problem formulation as well as solutions and approaches to the problems defined in these works. Section 4.3 discusses the hybrid switch design, defining the central problem of the different assignments of flows to the electronic or the optical switching core of the switch. We explain in this section the uniqueness of the problem described in this chapter. Section 4.4 discusses the Integer Programming formulation, the inputs to the Integer Program, and the constraints. A discussion of the choices made in this formulation is also presented at length. Section 4.5 concludes the chapter.

4.2 Previous Work

Shaikh, Rexford and Shin [21] address the routing of long-lived IP flows and propose a method to allow dynamic routing of these flows while routing short-lived flows over pre-provisioned paths. This allows for a reasonable solution to load-sensitive routing that does not lead to flapping[4]. The work suggests the ability to distinguish between two different sets of flows to improve stability and performance. The distinction between the different flows is made by measurement, which is an issue of concern: measurement assumes the ability to predict flow characteristics based on previous or historical arrival patterns. That is not always the case, since traffic patterns change with the changing applications using the Internet in general. In contrast, the work presented here provides two levels of differentiation through two methods (high-priority and low-priority flows) but assumes a Service Level Agreement (SLA) that defines the flow priority.

[4] Flapping is described as the dramatic fluctuations in link state between successive update messages under large update periods relative to the arrival rates and holding times.

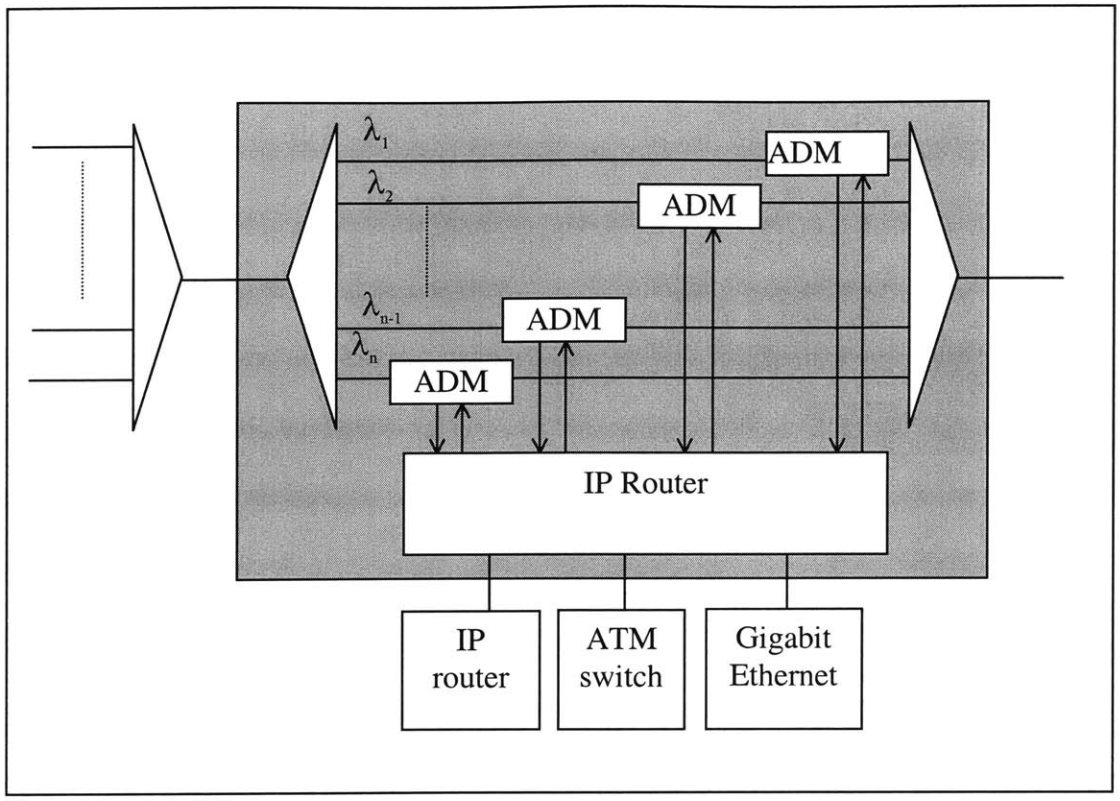


Figure 4-2 Component-Intensive Switching Process

This distinction between long-lived and short-lived flows develops from work done earlier by Newman, *et al* [22] that proposed a framework, which became the basis for Multi-Protocol Label Switching (MPLS) [6]. That work advocated the use of a switching function that would allow a node to make use of labels for certain flows set up in a fast forwarding table rather than try to locate the entry for each packet individually by searching the OSPF-based forwarding table. This concept of having different forwarding techniques will be used in this discussion where high-priority traffic will try to make use of OXC's and low-priority traffic will use mostly the IP router subsystem.

Banerjee and Mukherjee [23] present a more complete Integer Linear Programming formulation for wavelength-routed networks. Their approach takes into

consideration optical switching cores, and assumes the use of a number of wavelength converters. They allow the splitting of flows when solving the Integer Linear Program, an assumption that leads to packet reordering and bad performance at the transport layer. TCP does not deal well with reordered packets, leading the TCP protocol to go into congestion control and congestion avoidance, a behavior that leads to a low utilization of the theoretically available bandwidth. The Integer Programming formulation that is discussed in this chapter does not allow flow splitting. In addition, the main concern that motivates the work of Banerjee and Mukherjee is the static Routing and Wavelength Assignment (RWA), whereas the main objective of this dissertation is the distribution of flows both statically and dynamically over available network resources.

Recently, Kodialam and Lakshman [29] have proposed an algorithm for integrated dynamic routing of bandwidth guaranteed paths in IP over WDM networks. In the algorithm they present, the switching of LSP's takes into account the combined knowledge of resource and topology information in both the IP and optical layers. They discuss the need to find "good" paths to satisfy the requests for bandwidth. They define the measure of goodness as the selection of a path that permits as many future requests to be routed as possible.

The heuristic algorithm that we develop for the dynamic traffic accommodates future requests by load balancing incoming traffic across the network resources. The work in [29] also assumes *a priori* knowledge of the traffic patterns in terms of maximum bandwidth request and expected source/destination pairs. This allows a better approach to distributing the load in face of expected future traffic.

The algorithm described tries to accommodate as many requests without requiring *a priori* knowledge of the future requests as assumed in the reference. Without any

expected analysis of the traffic patterns and requests for bandwidth, the goodness claim is not justified. The paper [29] discusses ways of integrating priority information, but that priority applies to ingress and egress nodes, versus prioritizing the traffic itself. This allows the algorithm to make wavelength assignments more easily. It is important to note that prioritizing traffic is harder than node priority since the claim of selecting some ingress and egress nodes as more important only holds in particular instances. In real-life situations, all the ingress and egress nodes are crucial for the operation of the network (this is the motivation behind their deployment) and carry traffic that is sometimes important (serving a customer's database) and sometimes less important (carrying that customer's web browsing traffic). The paper [29] also claims that the need for an online algorithm is justified by the fact that if the network did static routing, then it would not expect future requests. Our approach, described in detail in Chapter 4 and Chapter 5, allows the network to perform both static and dynamic routing in order to better address that problem.

4.3 The Hybrid Switch Design

Optical switches are currently deployed in limited fashion for the long-haul networks. We propose a hybrid solution that capitalizes on the existing electronic switching cores as well as pure optical switches by making use of each switch's functionality and capability. The electronic switch is ideal for switching low priority flows, whereas the optical switch is ideal for high priority flows.

Figure 4-3 represents the hybrid switch. We define optical traffic as data arriving on a certain wavelength such as the data on wavelengths λ_1 and λ_2 of fibers 1 and 2. We define electronic traffic as traffic coming over an interface to other routing/switching technologies such as ATM, Gigabit Ethernet or IP routers. We define a lightpath as a

path between source and destination where all switching is done optically (also known as transparent switching). We define a semi-lightpath as an opaque path such as explained in [24]. This is a path that is formed of the association of more than one lightpath, where a switch or device on the path terminated the lightpath and started a new lightpath. Usually, semi-lightpaths allow a network operator to find routes between source and destination that do not use the same wavelength between the source-destination pair. Optical traffic coming upstream from other similar switches is distributed over both the electronic and optical cores. Electronic traffic is groomed with other low-priority traffic at the IP Router or with high-priority traffic at the OXC. The hybrid switch performs the dual function of switching data within the network as well as interfacing with other networks, in a setting such as a Metropolitan Area Network.

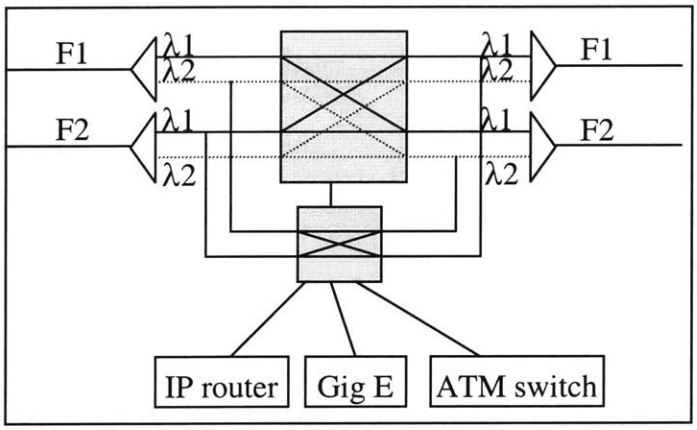


Figure 4-3 System Components in Hybrid Switching Model

Wavelengths that go through the hybrid switch are divided into high priority streams, ideally sent to the optical core, and low-priority data sent to the electronic switch. Low priority flows, such as web access, are better candidates to going through the electronic switch. Audio and Video streaming, IP telephony and large database

synchronization can use the lightpaths efficiently. An example of classification of high priority and low-priority flows is described in [21], with high priority traffic assumed to be long-lived flows that use a large amount of bandwidth and are set up for a long period of time, whereas low priority traffic could be mapped to short-lived flows that last for a limited amount of time and individually do represent a large load with respect to the network capacity. We assume that the priority of the flow is defined by either a Service Level Agreement (SLA) or by the network operator. This does not prevent the use of measurements to determine the flow quality or longevity, which could be the main metric used in the decision-making process. This work will assume knowledge of the flow priority throughout.

Each hybrid switch sends wavelengths to either the electronic switch, with the accompanying converter hardware, or to the optical cross-connect. The decision on the switching process is related to its prior switching upstream and the type of data it carries. For example, if a switch upstream used wavelengths λ_i and λ_j to aggregate low priority flows, then those wavelengths will be sent to the electronic switch at the current switch if possible. Data coming out of the electronic switch will be converted back to optical and multiplexed with outgoing wavelengths from the OXC. Data coming on the electronic interface includes high priority flows that would be better serviced by optical switching downstream, so the IP Router would forward them to the OXC. The OXC in turn will groom them to traffic in other existing wavelengths or send them in an unused wavelength, both of which would be switched optically downstream. This behavior applies to traffic that has been set up ahead of time, through the use of an optimizing function, as well as demand that is routed dynamically. We will discuss both approaches in Section 4.4 and Chapter 5.

4.3.1 Modeling the Hybrid Switch

The hybrid switch will be modeled by assigning costs to each of its parts. The approach allocates the different costs to the two types of flow. A high priority flow will incur a higher cost for being switched electronically, while a low priority flow will incur a higher cost by being switched optically. Figure 4-4 shows an instance of the hybrid switch model.

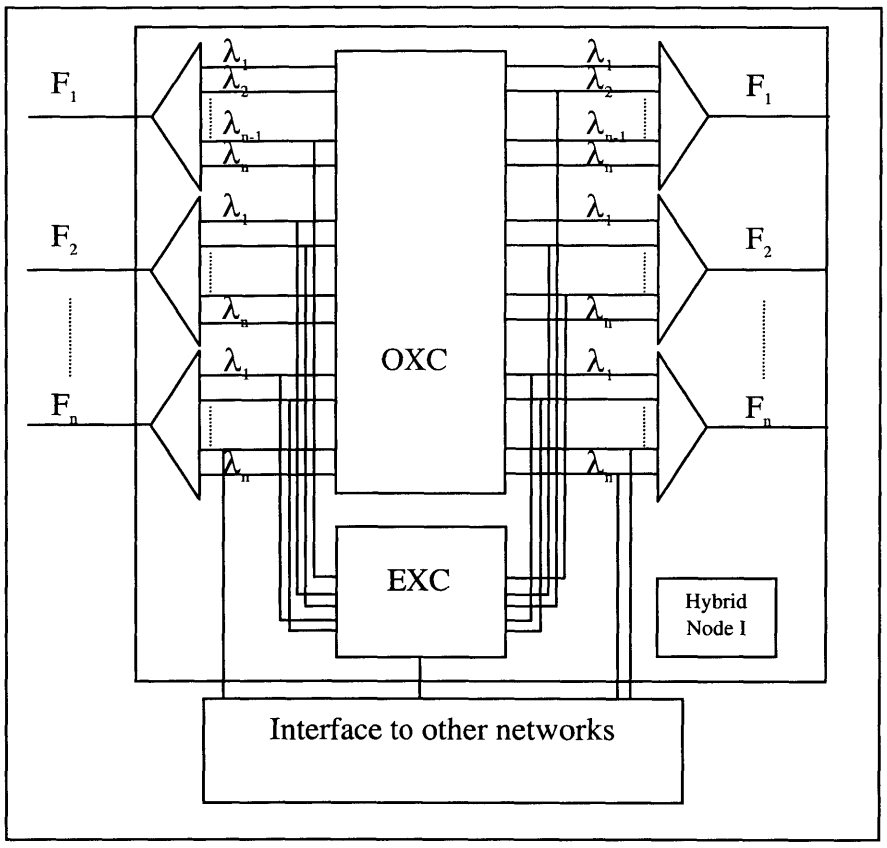


Figure 4-4 Model of the Hybrid Switch where some select wavelengths are terminated and sent to the IP router while the rest are switched in the optical cross-connect

The model shows both the components and the functionality of the switch. Fibers reaching the switch go through a demultiplexer to separate the different wavelengths. If one of the wavelengths λ has that switch as its destination, an Add-Drop Multiplexer

(ADM) function will terminate it by tuning its filter to λ; such is the case for λ_{n-1} of F_3. High-priority flows go ideally through the optical switching core, whereas wavelengths for low priority flows are terminated and the data they carry, sent to the electronic switching core.

For a node in the network, wavelengths λ_{n-1} of F_1, λ_1 and λ_2 of F_2, and λ_1 and λ_2 of F_3 are all switched electronically by having the wavelength filters tune to these wavelengths. The output wavelengths used downstream for these wavelengths are λ_2 of F_1, λ_{n-2} and λ_{n-1} of F_2, and λ_1 and λ_2 of F_3. The model also takes into consideration low-priority flows at the interface to be injected in the network by being groomed to the low-priority traffic through the IP Router. In addition, high-priority traffic could be added to outgoing fibers through the ADM function such as wavelength λ_{n-1} and λ_n of F_2. All wavelengths are multiplexed back and leave the switch through the outgoing fibers.

Table 4-1 Cost Assignment for Different Opto-Electrical Components of the Switch

	High priority flow	Low priority flow
OE converter	$c_{oe}(hp)$	$c_{oe}(lp) << c_{oe}(hp)$
EO converter	$c_{eo}(hp)$	$c_{eo}(lp) << c_{eo}(hp)$
Electronic switching function	$c_{es}(hp)$	$c_{es}(lp) << c_{es}(hp)$
Optical switching function	$c_{os}(hp)$	$c_{os}(lp) >> c_{os}(hp)$

Table 4-1 shows the costs associated with the different components. Making the analogy with OSPF routing, where data packets incur costs for traveling over links, our approach allows us to model the hybrid switch as a graph, and to optimize path selection.

The path cost within the switch will be the sum of costs of all links on the path as well as the cost of traversal of the switch components modeled as links. The graph transformation to allow for this will be discussed further. The cost of optical or electronic switching is infinite in the following cases:

- When a certain wavelength cannot be switched optically without wavelength conversion (N_o maximum wavelengths).
- When there are no more converters available.
- When the IP Router is fully utilized (N_e maximum wavelengths).

The cost of the components will be made a function of the utilization of the switch, when allocating paths and bandwidth dynamically. The utilization of the hybrid switch can be quantified as the fraction of total wavelengths that can go through the optical switch, and the percent utilization of the electronic switch. This is not the case for the static route assignment, where the network is solving for a set of demands and a given topology and bandwidth availability.

It is to be noted that all-optical wavelength conversion is currently impractical [25]. This means that were wavelength conversion necessary, in the model described above, the cost of wavelength conversion will be the sum of the costs of optical termination, OE conversion, then EO conversion, which has been described earlier. Wavelength converters as self-contained subsystems will not be considered in this discussion.

We will now discuss the routing and wavelength assignment for a set of demands, a given topology and a number of wavelengths per port. This will be defined as an Integer Program that can be run offline; the Integer Program can be run every specified amount of time to take into account new flows and re-optimize the whole network

routing.

4.4 Integer Program Description

We aim at solving the Wavelength and Routing Assignment by modeling it as an objective function and a set of constraints. For a certain demand matrix, our objective is to optimize path selection. This optimization problem deals with:

1. Minimizing the sum of the costs of all paths.
2. Minimizing the number of high priority flows that are sent in electronic switching cores.
3. Minimizing the number of low priority flows that are switched in the optical core.
4. Reducing the utilization of the electronic core (by reducing the number of high-priority flows that go through the electronic core).
5. Reducing the utilization of the optical core (by ensuring that not all the flows are switched optically).

Some of these objectives are conflicting ones, where we are trying to increase the number of high priority flows that are optically switched, but are also trying to free up wavelengths in the optical core. The intuition behind freeing up wavelengths is to make them available for future flow requests. The last two objectives are important to allow space for new flows to be set up. As mentioned above, the flows are differentiated based on their priority, making that a multi-commodity flow problem. In addition, the first objective of minimizing the sum of costs of all paths will allow us to achieve loop-free routing.

4.4.1 Definitions

We formally define all variables in Table 4-2.

Table 4-2 Definition and Description of Different Variables of Interest for the Integer Programming Formulation

f	Flow and its type: low or high priority; flows are characterized by their source, destination and priority
F	Set of all flows
b_f	Share of bandwidth needed by flow f
λ	Wavelength
Λ	Total wavelengths available
sw	Switch
SW	Set of switches
p	Port on switch
P^{sw}	Set of ports of switch
$(p_i:p_j)$	Fiber connecting port i of switch m to port j of switch n
P_{in}^{sw}	Set of incoming ports of switch sw
P_{out}^{sw}	Set of outgoing ports of switch sw
$y_\lambda^{p,sw}$	$\in \{0, 1\}$. 1 if use of wavelength λ
$x_\lambda^{p,f,sw}$	$\in \{0, 1\}$. 1 if use of wavelength λ for flow f on port p of switch sw
$z_\lambda^{p,sw}$	$\in \{0, 1\}$. Make use of wavelength λ for electronic (0) or optical (1) switching
B_λ	Bandwidth of wavelength λ
$c_{oe}^{f,sw}$	Cost of optical-electronic conversion for flow f. This cost is 0 if the traffic is coming on the interface to the local network, as in switch sw is

	the originator of the flow: sw=s^s
c_{es}^{f}	Cost of switching flow f through the electronic core
$c_{eo}^{f,sw}$	Cost of electronic-optical conversion for flow f. This cost is 0 if the traffic is leaving on the interface to the local network, as in switch sw is the terminator of the flow: sw=s^t
c_{os}^{f}	Cost of switching flow f through the optical core
$c_{l}^{p,f}$	Cost of using an outgoing link on port p for flow f (can be the same for both types of flows or can be differentiated based on priority; in this text, we assume different cost for different types)

The inputs to the problem are: f, F, b_f, $(p_i:p_j)$, P_{in}^{sw}, P_{out}^{sw}, Λ, SW, B_λ, $c_{oe}^{f,sw}$, c_{es}^{f}, $c_{eo}^{f,sw}$, c_{os}^{f} and $c_{l}^{p,f}$. The results of the IP will be $x_\lambda^{p,f,sw}$, $y_\lambda^{p,sw}$ and $z_\lambda^{p,sw}$. We have defined in Table 4-1 the cost assignments for the different components of the switch, depending on the nature of the flow going through such components. Note that the different costs of optical switching, electronic switching and OE conversion obviously are not all used for one flow *f*. This is taken into account by adjusting the cost according to whether the flow makes use of the converter(s) and/or one of the switching cores. For example, a flow going through the electronic switch will incur the OE conversion cost, the electronic switching cost as well as EO conversion cost, whereas a flow coming on the optical layer and switched all-optically will only incur the optical switching cost. Since the objective function is also a function of the number of wavelengths utilized, the optimizer will attempt to groom flows.

4.4.2 Problem Formulation

Since flows are assumed to belong to two classes of flow, looking at multi-commodity flow problems is important. This formulation is inspired by work presented in [26] and

in [23], but takes into consideration the optical-electronic hybrid model described above. An IP formulation of the objective function to solve this problem is as follows.

$$Minimize \sum_{sw\in SW}\sum_{p\in P}\sum_{\lambda\in\Lambda}\left\{ y_\lambda^{p,sw} + \sum_{f\in F} x_\lambda^{p,f,sw} b^f \left[c_l^{p,f} + \left(1 - z_\lambda^{p,sw}\right)\left(c_{oe}^{f,sw} + c_{es}^{f} + c_{eo}^{f,sw}\right) + z_\lambda^{p,sw} c_{os}^{f} \right] \right\}$$

Subject to:

$$x_\lambda^{p,f,sw} \in \{0,1\} \tag{4.1}$$

$$y_\lambda^{p,sw} \in \{0,1\} \tag{4.2}$$

$$z_\lambda^{p,sw} \in \{0,1\} \tag{4.3}$$

$$\sum_{f}^{sw} b_f x_\lambda^{p,f,sw} \le y_\lambda \le 1, \forall\lambda \in \Lambda \tag{4.4}$$

$$\sum y_{wl} \le N_o + N_e \tag{4.5}$$

$$N_o + N_e = \Lambda \tag{4.6}$$

$$P_{in}^{sw} + P_{out}^{sw} = P^{sw} \tag{4.7}$$

$$\begin{aligned} &\text{if } sw = s^s, \text{then} \sum_{p\in P_{in}^{sw}} x_\lambda^{p,f,sw} = 0 \text{ and } \sum_{p\in P_{out}^{sw}} x_\lambda^{p,f,sw} = 1, \\ &\text{if } sw = s^t, \text{then} \sum_{p\in P_{in}^{sw}} x_\lambda^{p,f,sw} = 1 \text{ and } \sum_{p\in P_{out}^{sw}} x_\lambda^{p,f,sw} = 0 \\ &\text{if } sw \notin \{s^s, s^t\}, \text{then} \sum_{\lambda}\sum_{p\in P_{in}^{sw}} x_\lambda^{p,f,sw} = \sum_{\lambda}\sum_{p\in P_{out}^{sw}} x_\lambda^{p,f,sw} \end{aligned} \tag{4.8}$$

$$\sum_{\lambda\in\Lambda} x_\lambda^{p_1,f,sw_1} = \sum_{\lambda\in\Lambda} x_\lambda^{p_2,f,sw_2} \text{ for each } (p_1 : p_2) \tag{4.9}$$

$$\sum_{p\in P_{in}^{sw}}^{sw} z_\lambda x_\lambda^{p,f,sw} = \sum_{p\in P_{out}^{sw}}^{sw} z_\lambda x_\lambda^{p,f,sw} \tag{4.10}$$

$$z_\lambda^p \left(x_\lambda^{p,f_1,sw} - x_\lambda^{p,f_2,sw} \right) = z_\lambda^q \left(x_\lambda^{q,f_1,sw} - x_\lambda^{q,f_2,sw} \right) \tag{4.11}$$

Let us now explain the objective function itself. The objective function considers a switch as the element of interest. The element $y_\lambda^{p,sw}$ adds to the number of wavelengths used; therefore using it in the objective function will ensure that we reduce the utilization of the optical core, which is one of our stated aims in the problem formulation. The sum over all flows f takes into consideration either the flow going through the optical switching core or the flow going through the electronic switching core. We multiply that by $x_\lambda^{p,f,sw}$ to take into account the fact that we use that wavelength for that flow. In addition, since the use of equipment is proportional to the fraction of total wavelength bit-rate B_λ, then we multiply the costs by the fraction of bandwidth b_f of the flow. This allows the grooming capability to add a certain fraction of cost on the overall switch function. This is done by multiplying the variable $z_\lambda^{p,sw}$, a Boolean variable that takes the values of 0 and 1 if the switching is done electronically or optically respectively, by the cost of going through either core. We also take into account the cost of the link on path p for flow f; this removes any routing loops that would have developed otherwise.

- If the switching is done optically, then we need to account for the cost of that switching c_{os}^f and the cost of using the outgoing link on a certain port for that particular type of flow. This allows us to minimize the sum of the costs of all paths. In addition, the value we set c_{os}^f at, as described in Table 4-1, based on flow priority, ensures that more high-priority flows go through the optical switch.
- If the switching is done electronically, then the cost of equipment use is proportional to $c_{oe}^{f,sw} + c_{es}^f + c_{eo}^{f,sw}$ (going through optical-electronic converters if it does not originate at *sw*, going through the electronic core, then conversion back to optical if it does not terminate at switch *sw*). Most probably, several flows will

be aggregated on this low-priority wavelength; this supports the approach of making the cost of using a flow's wavelength proportional to the total bandwidth that wavelength can carry.

Costs are summed for all wavelengths, over all ports and for all switches.

Constraints (4.1), (4.2) and (4.3) indicate that the variables are integer variables that can take the values 0 and 1 only. Constraint (4.4) enforces that the sum of all flow bandwidths over a certain wavelength be less than one; it has to be zero for other wavelengths. Constraint (4.5) says that the total number of wavelengths used in a switch cannot be greater than the total sum of all wavelengths that can be switched optically and the wavelengths that can be switched electronically. Constraint (4.6) sets that total number to Λ. Constraint (4.7) sets the total number of ports of a switch to the sum of input ports and output ports on that switch.

Constraint (4.8) deals with flow conservation. If the switch is the source for flow f, then the sum over all input ports of fractions of flow f is equal to 0 (flow f not coming on any input port), whereas the sum over output ports is 1 (flow f leaving in its entirety on an output port). The reverse is true if the switch terminates flow f, then the sum is 1 over all input ports and 0 over all output ports. If the switch is neither source nor destination of the flow f, then this flow coming on all input ports for all wavelengths is equal to the sum of all parts of the flow over the output ports.

Constraint (4.9) deals with the routing problem, by enforcing that a flow f leaving switch sw_1 on output port p_i to go to a switch sw_2 on output port p_j must use the link (p_i-p_j).

Constraint (4.10) enforces that if a flow f comes at a switch on an input port p on a wavelength λ and is switched optically, it should leave that switch on the same

wavelength λ, thereby ensuring that the same wavelength is used and no wavelength conversion occurs.

Constraint (4.11) ensures that flows f_1 and f_2 coming on a wavelength λ with both leaving on the same wavelength λ are switched using the optical switching. This enforces that the solver does not perform some load balancing by splitting flows of an optically switched wavelength into two other wavelengths. It is physically impossible to split flows optically today. By inspecting this equation, several cases are studied. If the left-hand side of the equation is 0, then two cases arise:

1. The wavelength is switched electronically: z on both sides of the equation is 0 (equality is verified).
2. The wavelength is switched optically: then both flows are on the wavelength. To verify the equality, at the output of the switch, they should be on that same wavelength.

If the left-hand side of the equation is equal to 1, then one case is considered:

- Flow f_1 comes on port p, on an optically switched wavelength λ, but f_2 is not on wavelength λ or port p. f_1 will go out on the same wavelength at output port q, but f_2 will not.

If the left-hand side of the equation is equal to -1, then the reciprocal case applies.

- Flow f_2 comes on port p, on an optically switched wavelength λ, but f_1 is not on wavelength λ or port p. f_2 will go out on the same wavelength at output port q, but f_1 will not do the same.

Those constraints conclude the presentation of the Integer Program.

4.4.3 Discussion on Integer Programming Formulation

The Integer Programming formulation is a global solution to the defined problem. While

this solution allows for a defined set of flow demands to be routed optimally on a certain topology, we would like to explore more dynamic approaches in Chapter 5 that can do local forwarding. New flow requests make the prior setups sub-optimal with respect to the Integer Programming formulation. On the other hand, solving the Integer Program would be impractical if it were done on a per-flow basis, because of the length of time required, and because rerouting all the flows would lead to a bad network performance. Therefore, we make use of it to set up the paths initially, and make use of a dynamic decision-making algorithm to set up new flows. That algorithm will handle incoming flow requests by assigning them to paths with sufficient bandwidth from the set of available paths. Then, at longer time intervals, the IP would be solved again to re-optimize the traffic streams. The combination of static wavelength allocations and dynamic routing allows the network operator to increase the network usage while optimizing traffic routing over longer intervals.

4.5 Conclusion

We have developed an Integer Programming formulation for the static routing and wavelength assignment using a hybrid optical and electronic switch. The Integer Program allows us to solve the problem for long-term requests as opposed to dynamic demand. This allows us to optimize the network resources and availability for future requests as well. Lessons learned in the optimization problem allow us to devise a heuristic algorithm to be implemented for dynamic requests, the subject of Chapter 5. Simulation of the heuristic algorithm allows us to gather data that infer the appropriateness of the algorithm and its ability to increase network utilization and load balancing.

Chapter 5

Heuristic Algorithm for Hybrid Packet and Circuit Switching

5.1 Introduction

The heuristic algorithm's objective is not to deliver optimal routing of all traffic; rather, it aims at allowing an intelligent dynamic admission process. This ensures a quick admission process that does not negatively impact already-admitted flows.

A hybrid switch makes local decisions as to whether to terminate a wavelength, and switch data electronically or switch that wavelength optically. If a switch starts a new wavelength at the Add-Drop Multiplexer (ADM), but that wavelength is used downstream by a low-priority flow, then even if the new flow were a high priority flow, the switch downstream would have to terminate it, reducing the usefulness of this

assignment. This is especially true in the case of a flow coming from other networks such as the IP, Ethernet and ATM networks depicted in Figure 4-3. Section 5.2 presents the requirements for the heuristic algorithm that allows us to add flows dynamically without running the Integer Program on a per-flow basis. Section 5.3 discusses the ability and need to build eligible paths prior to accepting requests, to increase the speed of call admissions. Sections 5.4 and 5.5 describe the algorithm and how it behaves depending on where the traffic request is coming from (optical or electronic interface). Section 5.6 is an example running of the algorithm. Section 5.7 describes our testing methodology with a brief discussion on network topology as well as the simulation environment. Sections 5.8 and 5.9 present the simulation results and how we can interpret them.

5.2 Requirements of The Heuristic Algorithm

Figure 5-1 shows an example of such an assignment. In this example, nodes N-1 and N switch wavelengths λ_2 of fiber F1 and λ_2 of F2 optically. At node N-1, input wavelength λ_1 of fiber F1 is unused. Wavelength λ_1 of fiber F2 is switched electronically, with more traffic added to it through the electronic interface. A new high-priority flow is assigned the outgoing wavelength λ_1 of fiber F1. That wavelength though cannot be switched optically at node N to wavelength λ_1 of outgoing fiber F2 because that wavelength is used for low-priority traffic coming in on the electronic interface. In that case, another route through other switches to the sought destination may allow all-optical switching of that high priority flow. Therefore, wavelength assignment decisions at a certain node should be made with knowledge of the wavelength availability at other switches in the network.

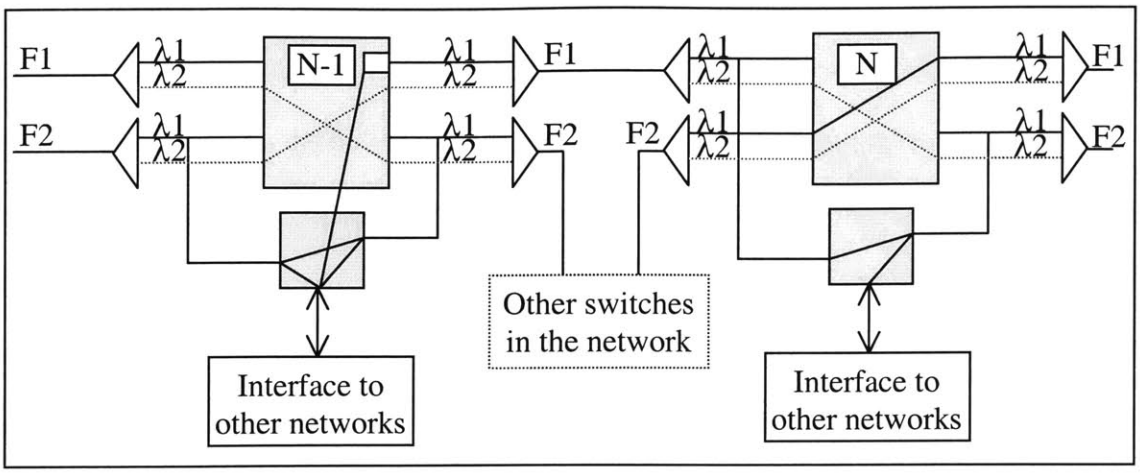


Figure 5-1 Problems with Wavelength Assignments

When new traffic flows are added, the link costs will change according to the utilization of the different components of the switch. As mentioned earlier, work has been done to approximate the General Routing Problem's optimality by using the conventional OSPF routing mechanism but making intelligent link cost assignments. Link cost assignments will be performed using the cost-setting work described in [15]. This allows for a distribution of the network load. Nodes only set up wavelengths based on availability. The local decision-making process reduces the probability of crankback[5] [27], a situation where a node makes its flow-routing decision based on old information but the nodes downstream cannot accommodate that flow.

A hybrid switch keeps its state updated and advertises the new costs associated with each of its wavelengths. Different costs are maintained for both flow types as

[5] In this situation, since a node makes routing decisions based on outdated information, nodes that were identified as downstream nodes and thought to have enough capacity to carry the traffic may in fact not be able to do so. This leads to a situation where the path setup requests have to be detoured around the blocked node that does not have sufficient resources. This leads to the creation of sub-optimal paths that use unnecessary resources in the network.

mentioned previously. Costs are a function of the priority and utilization of the wavelength. For a wavelength geared for optical switching, the cost for high-priority flows is smaller than that for low-priority flows. The same advertisement happens when a flow no longer goes through the hybrid switch. This happens when the switch is notified of the change when a Label Switched Path (LSP) is torn down or when a flow reservation is idle for a long time, leading to its teardown. The cost is updated to reflect the new cost for a flow to go through the switch and the resources it will utilize. For example, for a new high priority flow, if the switch still has wavelengths available for optical switching, the cost as seen by neighboring switches will be $c_{os}(hp)$, whereas for a low priority flow coming on the electronic interface, the cost will be $c_{oe}(lp) + c_{es}(lp) + c_{eo}(lp)$. If the switch upstream is an IP Router (traffic coming on the electronic interface), then the cost for a high priority flow will be $c_{es}(hp) + c_{eo}(hp)$ whereas the cost for low priority flows would be $c_{es}(lp) + c_{eo}(lp)$. The switch advertises the new costs in addition to the available wavelengths on each of the outgoing ports and each wavelength's remaining bandwidth. This broadcast mechanism is important for the following reasons:

1. The switch will exist in a heterogeneous environment with other switches not supporting that cost assignment and structure.
2. Physical changes made to the switch (adding more ADM's for example) should not be advertised to the rest of the network.
3. A switch can estimate its own utilization accurately.
4. When making forwarding decisions, the switches upstream at the edge of the network would make a path selection based on the utilization/congestion of the network links and switches downstream.
5. The list of wavelengths available allows other switches to make an informed

decision on what wavelength to pick for high-priority flows to ensure as transparent (all-optical) switching from source to destination as possible.

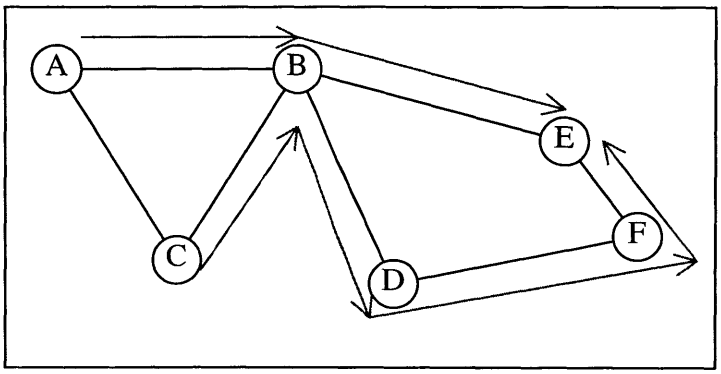


Figure 5-2 Behavior of The Heuristic Algorithm

Figure 5-2 shows the behavior of the heuristic algorithm in a simple network topology. Node A wants to set up a new high-priority flow to destination E that uses the total bandwidth that a wavelength can carry in this network. We assume two wavelengths per fiber in this example, one geared for high-priority flows, and the other for low-priority flows. Costs for crossing each switch are set at 0 for optically switching high-priority flows and 10 for the electrical switching for all ports. Costs per fiber link are assumed constant at 1. The algorithm will perform the wavelength assignment on path (A-B-E). Node B will update the network about the new costs of crossing it, which become infinity for optical switching on ports B-E and B-A but stays constant for electronic switching. Before the update reaches other nodes in the network, a new high-priority flow request at node C has destination E. C is unaware of the change in costs at B and its shortest path tree shows the path to be the optically switched lightpath (C-B-E), so it forwards data on the wavelength towards B. Node B has no optical switching capacity and the outgoing

wavelength (B-E) is geared for low-priority flows, so node B's first action is to terminate the incoming wavelength at a cost of 10. B finds a new shortest path to E that consists of lightpath (B-D-F-E) of cost 0+1+0+1+0+1 (going through 3 OXC's by using arcs B-D, D-F and F-E. The same applies to D and F that will forward the data by using their all-optical switching capability. So, both a view of the entire network and local decision-making allowed us to find paths to the destinations requested by the incoming flows.

In the case of switches that do not support this process, they will be assigned automatically a constant cost of switching. This is done to ensure that the path choice is not made to go through older switches that appear to add 0 to the cost function, as opposed to the hybrid switch that is adding a certain cost of passage. The network operator could also assign a cost as a function of the capacity of that switch. This allows for a low deployment cost by not putting the restriction of upgrading all switches at the same time. In addition, one of the major concerns of carriers is the downtime of their networks. Forcing a complete overhaul of the network would definitely incur a large downtime that is not affordable by the carriers because of their Service Level Agreements (SLA's) with their customers.

5.3 Building Eligible Paths Prior to Accepting Requests

Each switch maintains two shortest path trees that it uses, one for the low-priority traffic and the other for the high-priority traffic. Figure 5-3 shows the pseudo-code for the pre-computation at all nodes.

```
At every node N:

- Apply graph transformation on the graph to build a dual
graph that transforms node components into arcs of the dual
graph
- Calculate node and arc costs using method described in
reference [5]
```

```
- Calculate Shortest Path Tree to all destinations using IP
Router
- Prune all arcs with capacity = 0
- For ∀ λ ∈ Λ, calculate SPT
```

Figure 5-3 Path Pre-computation at All Nodes

The first part of the computation allows the algorithm to take into account the cost of going through a node. A straightforward way is to build a dual graph by replacing every node with a set of intermediate arcs and nodes connecting incoming and outgoing links, through a central node. Figure 5-4 shows such an example, where node 1 is replaced by nodes 1_e, 1_o (two optical wavelengths are replaced by two nodes; this allows us to build multiple lightpaths based on the number of wavelengths available), 1', 1", and 1'". This allows the algorithm to run an un-modified shortest path algorithm such as Dijkstra. We take into account different component costs as advanced earlier, by setting the arc costs at node n as shown in Table 5-1.

Table 5-1 Costs for Arcs Inside Hybrid Switching Nodes

Arc	Cost
$(n^i;n_e)$	$c_{oe} + c_{oe}$
$(n_e;n^i)$	c_{eo}
$(n^i;n_o)$	c_{os}
$(n_o;n^i)$	0

These costs are of course dependent on the commodity that crosses them. To explain this cost allocation, let us look at the different cases that may arise:

- A traffic stream goes from j to k through 1 and is switched optically, which implies that it goes through j, 1', 1_o, 1" and k, therefore incurring in node 1 the cost: c (1'- 1_o) + c (1_o- c1") = c_{os}.

- A traffic stream goes from j to k through 1 and is switched electronically, which implies that it goes through j, 1', 1_e, 1" and k, therefore incurring in node 1 the cost: c (1'- 1_e) + c (1_e- c1") = $c_{oe} + c_{es} + c_{eo}$.
- A traffic stream is received at the electronic interface at node 1 going to k through 1, which implies that it goes through 1_e, 1" and k, therefore incurring in node 1 the cost: c (1_e- c1") = c_{eo}.

Two piecewise linear increasing and convex functions account for link costs based on utilization, one for high-priority flows and the other for low-priority flows. We assume that all nodes are equivalent, therefore, all link costs (1^i- 1_e) and (1_e- 1^i) are based on the same function, and all link costs (1^i- 1_o) and (1_o- 1^i) are equivalent. All links between nodes are assumed to have a cost of 1.

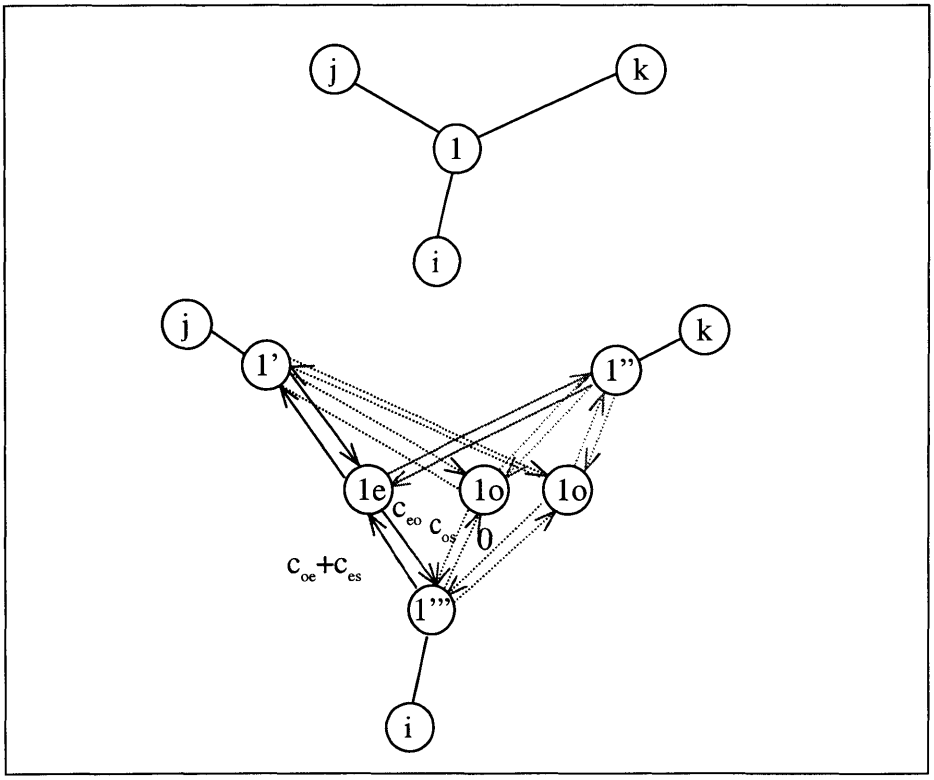


Figure 5-4 Graph Transformation Takes Node Cost in Account

The second part of the computation is to calculate the Shortest Path Trees (SPT) of interest. For low-priority traffic, the SPT will determine shortest paths by considering the IP Router and the wavelengths assigned to low-priority traffic in Section 4.4. For high-priority traffic, it will build based on each wavelength the available SPT. Since some wavelengths may have been already fully utilized, some paths cannot be used for all-optical switching. In this instance, a semi-lightpath may have to be set up while taking into consideration the extra cost of electronic switching.

5.4 Traffic Request at Electronic Interface

When a node receives a new flow request coming on the electronic interface, if the request is for high-priority traffic, the node will traverse the appropriate shortest path tree to assign a (or reuse an existing) wavelength. It is also important to keep the wavelength conversion (through electronic switching) or wavelength termination to a minimum. For a flow that would have to be terminated when using the shortest path, the cost of termination obviously increases the total cost of the path. Once a path is selected as the smallest-cost path, the node makes sure that the available bandwidth on the assigned wavelength is greater than that of the requesting flow. If no wavelength conversion is needed, then the node will set up that wavelength (using possibly Optical Burst Switching, and indicating its priority to the node downstream) if it wasn't used previously. If the node were adding the flow to an existing wavelength, then it would just update the utilization of that wavelength. If the node cannot find a non-terminating lightpath, then it will compare the total path cost to that of the cost of using a wavelength using the other short paths if any exist. The path with smallest cost will be used. If the new flow is low-priority, then the shortest path will be used out of the shortest-path tree exclusively. Since it is the goal of this work to keep optical switching for high-priority

flows, it is not necessary to compare the total cost to those of other paths obtained. Figure 5-5 shows the pseudo-code for this section. The algorithm forms semi-lightpaths iteratively by switching wavelengths electronically when optical switching cannot occur. It makes decisions based on whether the flow request is coming on the electronic interface or the optical interface, the available bandwidth, the flow priority and the possible lightpaths that can be set up.

```
If request on electronic interface {
    If high-priority flow {
        If SPT has enough bandwidth on a lightpath to
destination
            Accept flow on that path and notify node
downstream
        Else
            Find another short path with no routing loop
that can sustain the new flow
        Else
            Search in SPT's for lightpath and accept flow on
shortest found path if any
        Else
            Find lowest cost semi-lightpath
    }
    Else
        Accept traffic in IP Router if bandwidth available
}

Find_lowest_cost_semi-lightpath() {
    For original SPT, path to destination has certain cost
    For every λ, for that SPT {
        Every time the traffic stops to destination (no more
bandwidth or no wavelength),
        Add cost of electronic switching to new wavelength
downstream
        Calculate total path cost
    }
    Pick smallest cost path
}

If request on optical interface {
    If high-priority flow {
        If next hop in SPT can be reached by optical
switching of that wavelength
            Switch wavelength optically to that hop
        Else if other path for that wavelength can be found
            Switch wavelength optically to port for next hop
        Else {
```

```
            Find wavelength for new lightpath to destination
            Switch electronically to that wavelength
    }
    Else
        Switch electronically
}
```

Figure 5-5 Pseudo-code for New Traffic Flow

5.5 New Traffic Flow at Optical Interface

The OXC decides to switch a new incoming data flow between two different optical ports by performing the following operations. Flows come with control data either embedded in the case of market traffic packets or out-of-band in the control channel. If the control data it receives about that wavelength defines it as a high-priority flow (low priority), the switch will attempt first to forward it downstream using the optical (electronic) switching core. It will use the same decision-making process presented earlier. The cost of OEO conversion is taken into account when making the choice of path by adding it as a cost to the shortest path tree for the high-priority traffic (low priority traffic). This means that the cost of conversion through the switch is taken into account if the wavelength assignment downstream is different from that upstream. That cost is always dependent on the nature of the traffic.

Once a path is assigned, the node will recalculate its right-of-passage cost by increasing/decreasing costs based on utilization and publicize the following:

- Node cost for high-priority flows.
- Node cost for low priority flows.
- Available unused wavelengths.
- Used wavelengths' utilization for the wavelengths that are started at this node (since that node can inject data in these wavelengths that may be optically switched downstream).

These costs would change based on whether the electronic (optical) core is heavily utilized and whether it can switch more wavelengths electronically (optically). It is assumed that low-priority traffic has less resource needs than high-priority traffic. All nodes receiving this control information including the node itself will update their shortest path trees. We can achieve this by running an algorithm such as the dynamic SPT algorithm discussed in [28]. This allows us to save on the cost of re-running a Dijkstra algorithm. The nodes will also build a new list of short lightpaths to account for the changing node costs; they will update the paths with the appropriate wavelength usage on each of the links.

5.6 Example Running of the Heuristic Algorithm

To describe the heuristic algorithm, we show an example in Figure 5-6. Graph G = (N, A) is the original network topology and G' = (N', A') is the transformed graph taking into account the node costs. For example, a flow that goes on path A-B-C travels on A'-B'-Bo-B"-C", thus taking into account the cost of going through switch B.

In this example, we assume that the cost setting function for OSPF-TE [15] will change the link costs as described in the scenario shown in Table 5-2. The table shows flow requests that come sequentially to the network. After every new request is fulfilled, arc costs on the transformed graph are recalculated to account for the added load. We assume that some paths have already been set up and that the remaining link bandwidth consists of 3 wavelengths per link, 2 of which are high-priority and 1 is for low priority traffic. In addition, we assume that each wavelength can carry 10 Mbps.

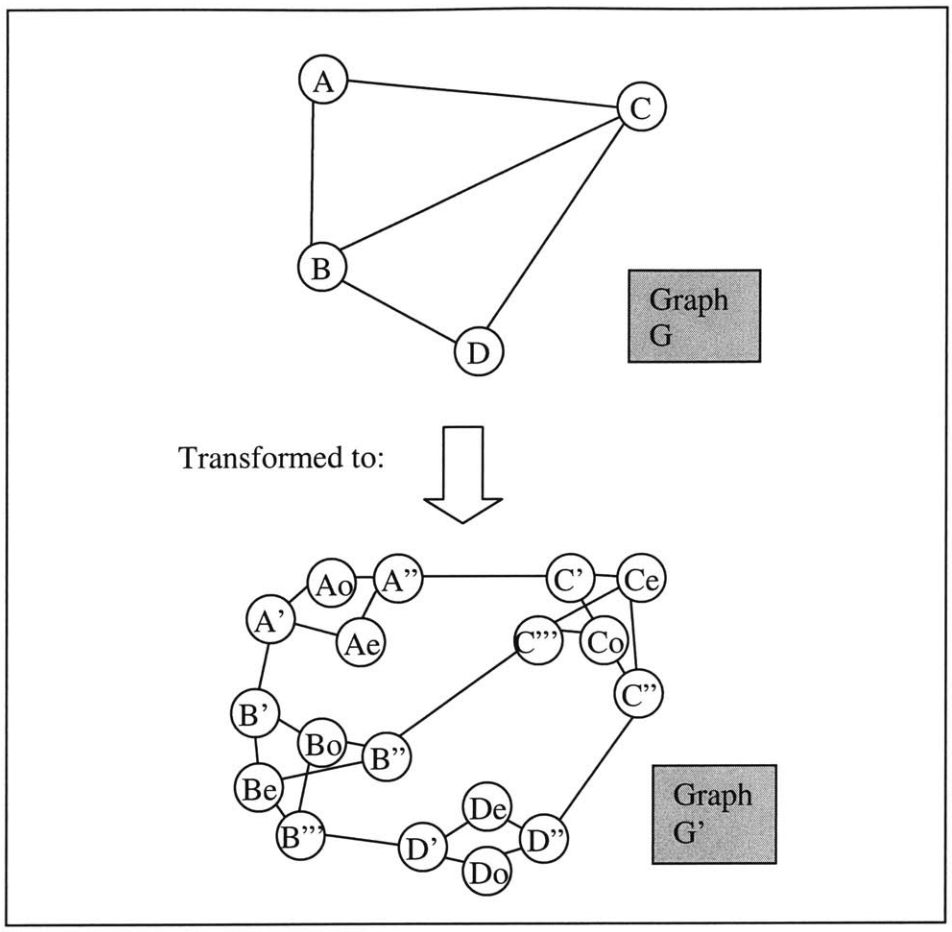


Figure 5-6 Example Application of the Heuristic Algorithm to Simple Network

The graph is an undirected graph. A request will be rejected if the flow has to be split to reach the destination. Our observations about the path decision are explained in the table. Cost changes sometimes do not affect the Call Admission Control process since the shortest path does not have enough bandwidth to fulfill the call. Such is the case of flow request 6 that cannot be switched optically at B to reach D (all high-priority wavelengths on AB are terminated at B). On the other hand, this plays a role for flow request 9. Arc D-B's low priority wavelength ($\lambda 3$) is used to transport traffic for flows 5 and 8; this leaves 2 Mbps to be used. In addition, the electrical switch at D is switching 8

Mbps worth of traffic. This increases the cost of electrical switching at D, therefore, arcs D'-De and De-D'' become high cost arcs on the transformed graph. In contrast, switch A is only switching electrically 3 Mbps for flow 5, with a smaller cost increase at links A'-Ae and Ae-A" in the transformed graph. The end switches C and B and the links (whether CA-AB or CD-DB) will go up in utilization in both cases; therefore, the heuristic algorithm sees a lower cost of going through A vs. D. It decides to use path C-A-B to fulfill that request. This has the effect of distributing the load on both the switches and the arcs.

Table 5-2 Flow Requests and Assignments

Flow request: Source Destination	**Priority**	**BW (Mbps)**	**Path Assigned (wavelengths)**	**Observations**
1: AB	High	8	A-B (λ1)	Uses λ1 on AB
2: AB	High	5	A-B (λ2)	Uses λ2 on AB
3: AD	High	4	A-C-D (λ1, λ1)	Cost for high-priority traffic of going through B is now higher than cost through C
4: AD	Low	3	A-B-D (λ3, λ3)	Electrical switching at B
5: AD	Low	3	A-C-D (λ3, λ3)	Electrical switching at C. The cost of going through B has increased, so A-C-D is a shorter path
6: AD	High	3	A-C-D (λ1, λ1)	A cannot reach D through B using optical core since both λ1 and λ2 on AB are terminated at B
7: AB	High	6	A-C-B (λ2, λ2)	Direct arc AB does not have the capacity to carry the traffic on any one wavelength (no splitting). If the flow goes on λ3 it will be switched electronically at B. If the flow goes

				on wavelength λ1 through C, it will be terminated too. So the flow goes through C on wavelength λ2 (least cost).
8: CD	Low	5	CD (λ3, λ3)	Uses low-priority λ3 on CD
9: CB	Low	1	C-A-B (λ3, λ3)	Uses low-priority λ3 on CA and AB. It doesn't go on C-D-B because the cost of electrical switching at D is higher (this is due to flows 5 and 8 that use 8/10th of available wavelength bandwidth, thus increasing the cost of switching)

5.7 Simulation Methodology

We discuss in this section the simulation methodology, explaining the choices made in building the cost and capacity matrices of interest, the running of the heuristic algorithm, implementation details about the simulation, and general problems we dealt with and their respective solutions.

5.7.1 Network Topology

The first crucial issue is to determine what network environment is most suitable for the simulation. In an ideal environment, synthetic topologies such as the ones discussed in [26] would be of instrumental value in determining the data points of interest to the simulation. Another topology that we considered is shown in [29], but that random topology shown in the research paper does not hold resemblance to any real network and assumes a heavily meshed network, as opposed to a sparser network, which is not the case of deployed networks. As the main objective of this work is to understand the impact of the hybrid switching system in a real environment, the choice of the topology needs to closely emulate a real-life network. The logic behind choosing a sparse network

in our case is because of the high fiber deployment costs due to several issues. The following is a non-exhaustive list, but one that nevertheless highlights the inappropriateness of using a random network topology.

- Regulatory issues: often overlooked, regulatory concerns from cities, towns and governments play a big role in the decision-making process on laying fibers; in smaller countries such as countries in Europe, several governments have to give approval of use of their land for communication purposes. They also want to have the right to closely monitor that trans-border information flow heavily, increasing the equipment cost and switching facilities.
- Rights of way: many optical fiber networks are built along highways to allow these networks to make use of a road architecture that changes little, while providing fast routes between different geographical locations. In addition, those highways are built to link areas with a large population density. The need for bandwidth between those locations is immediately obvious. On the other hand, highway authorities (both private and public bodies) will charge vast amounts of money to make use of that infrastructure, which limits the number of fibers that connect the nodes in the network.
- Ring architecture: While a lot of discussions have taken place about the inappropriateness of rings and the better network utilization that mesh topologies allow, the current networks are almost exclusively made with fiber rings in mind. One report in Light Reading online magazine mentions that one hundred thousand SONET rings are deployed today. This means that every node needs a minimal number of input and output fibers (though a much larger number of channels) and therefore the topology will be sparse.

- Other geographical problems such as the small number of bridges that connect different sites. Consider, for example, Boston to Cambridge, where three bridges permit land connectivity, forcing dependence on wireless connectivity. This shows that full optical wire-line mesh connectivity is typically not possible.

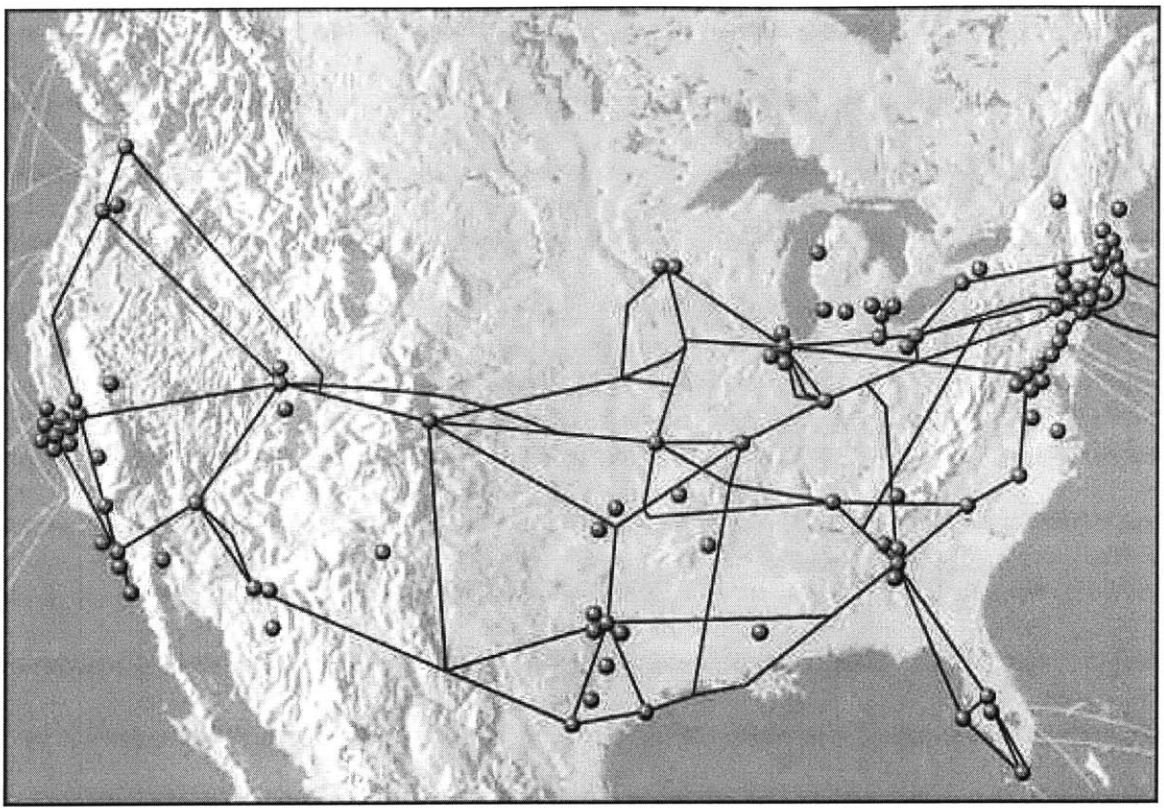


Figure 5-7 WorldCom's networks [40] depicting fiber optic networks, network facilities and international cable routes in 1997

Sample real topologies abound. For example, the topology shown in Figure 5-7 shows the network topology documented by MCI-WorldCom ® in its report, Building the Right Networks [40]. The figure depicts a sparsely connected graph, with switches at major metropolitan areas as expected. The number of nodes in this graph is the same order of magnitude as the number of links.

Another topology representative of a newer carrier clearly represents the thought

process in building rings to accommodate the protection and restoration needs required in today's SONET networks. Figure 5-8 shows the network being deployed by a newer carrier, Level 3 Communications ®, Inc. from scratch. A sparse mesh is very appropriate for this graph as well.

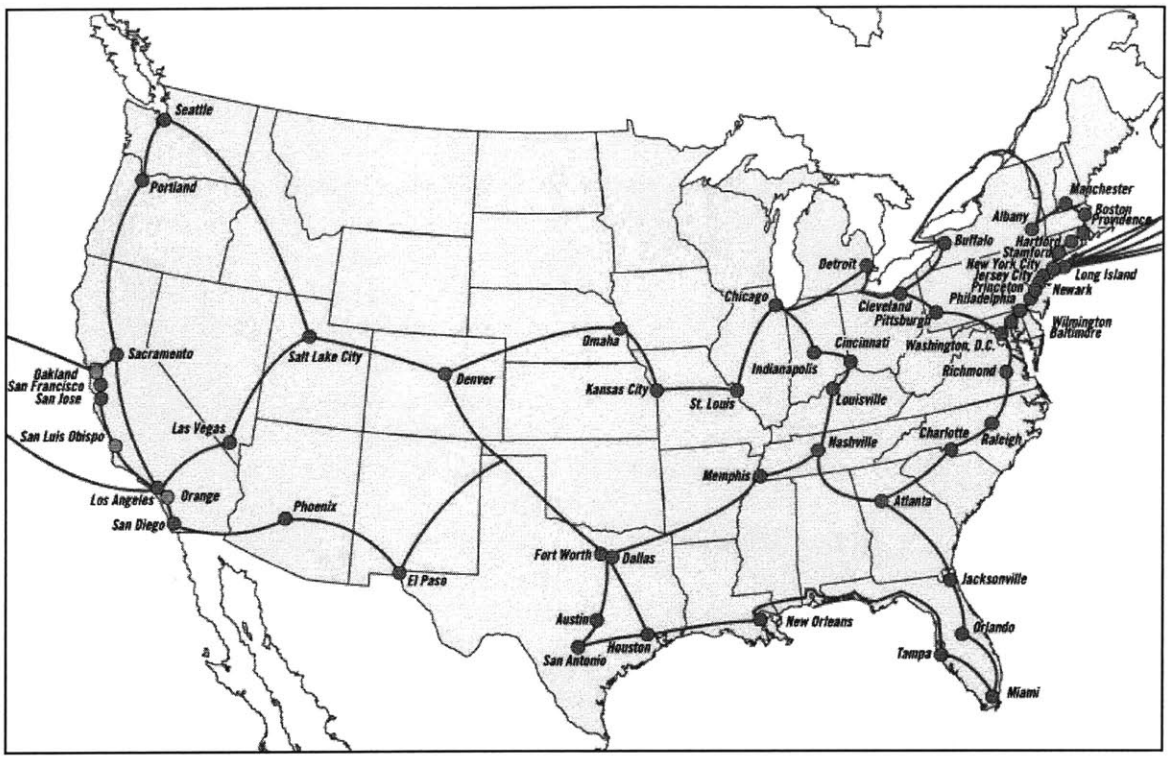


Figure 5-8 Level 3 Communications' network [41] depicting their fiber optic network using IP. Major switching nodes again reflect US demographics and urban development by being deployed in major metropolitan areas

This leads us to identify a topology of interest where meaningful data could be collected. This topology is based on the MCI topology, but for the sake of clarity, it has been redrawn to reflect only a select number of nodes and links of interest in a smaller geographic area that covers the US West Coast and parts of the Midwest. This topology serves our purpose well for identifying meaningful data.

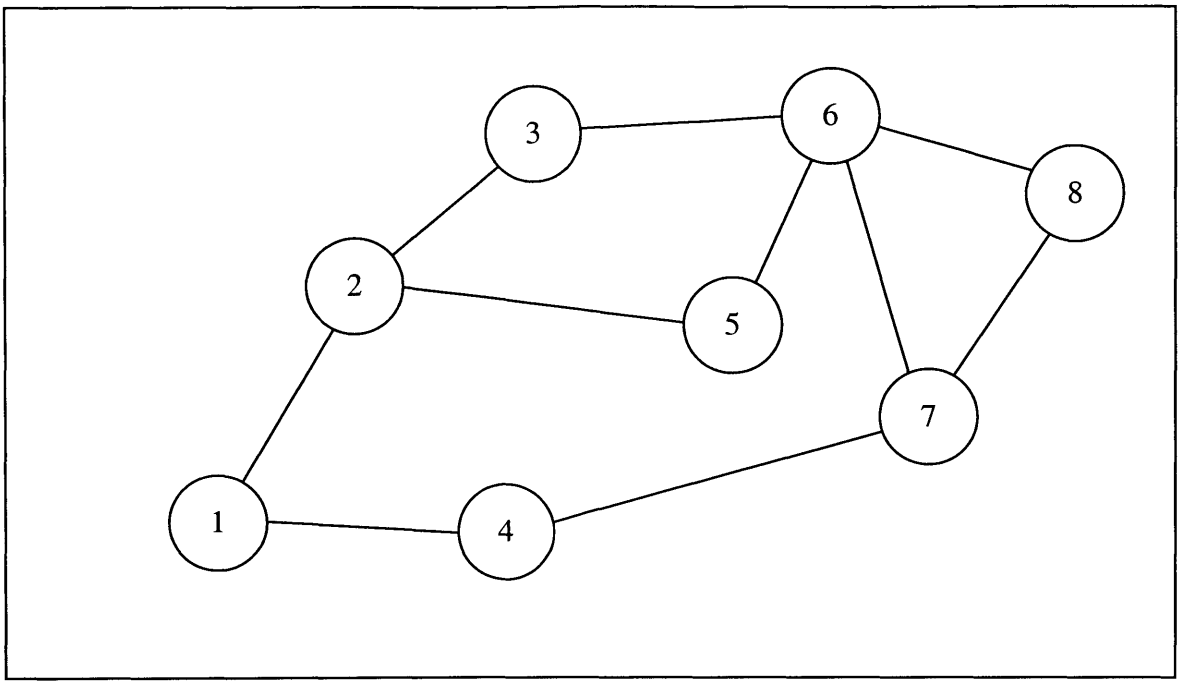


Figure 5-9 Network topology used in the simulation. All nodes are assumed to be hybrid switches

Figure 5-9 shows this network topology with all nodes assumed to be hybrid switches. In addition, all fibers connecting the nodes are bi-directional, meaning traffic can travel in either direction on each fiber strand. Modeling bi-directional fibers is done by replacing every fiber between any two nodes I and J with 2 directed links (I-J) and (J-I) of the same capacity.

5.7.2 Simulation Environment

We assume the number of wavelengths (channels) that each fiber carries to be eight, distributed by sending six to the optical cross-connect and two to the IP router. This closely emulates the assumption that optical switching cores allow the network to switch more bandwidth; a three-to-one ratio is realistic in this case. In addition, this can currently be deployed as a low-cost solution using CWDM (Coarse Wavelength Division

Multiplexing) that can use inexpensive un-cooled lasers and wide-band multiplexers and demultiplexers [45].

Each wavelength is assumed to carry bandwidth normalized to 1. The current state of the technology supports deployments at 10 Gbps per channel, while 40 Gbps per channel rates have been developed in several research labs and are currently being field-tested [46].

Since the algorithm being tested relies heavily on matrices, the choice of MATLAB ® for implementing the simulation was obvious [47]. For the OSPF protocol, Dijkstra's algorithm was used to find the shortest path routes. There are three matrices of interest:

- A node adjacency matrix that shows the links and capacities of these links between the different nodes and sub-nodes (each switched wavelength is assumed to go to a sub-node within the switch). This is a node-adjacency matrix that reflects the lightpaths that have been setup.
- Two cost matrices that represent the costs for all links seen by each of high-priority or low-priority flows. This is a node adjacency matrix that reflects the lightpaths that have been setup as well. The costs vary depending on loading of the link. The actual fiber links are assumed to carry infinite bandwidth; therefore fiber costs are set to one. This is in agreement with the piece-wise linear increasing and convex function that is used to calculate arc costs [15] as the arc cost increases as a function of the arc utilization (undefined when capacity is infinite). The piece-wise linear increasing and convex function allows the system to increase the cost of the link faster when the arc becomes more utilized.

A request consists of the tuple **(source S, destination D, capacity C)**.

When a high-priority traffic request comes in, the algorithm inspects the link matrix as well as the high-priority cost database. It checks that the node it is running on is not the destination itself. If it is indeed the destination of the flow, it sends the flow to the electronic interface. Otherwise, it prunes all link costs where **`capacity (link) < C`**. It then runs Dijkstra's algorithm to find the shortest path to the destination. If such a path is found (meaning the network can support the bandwidth requested), it will be used to route the traffic. That path may be a transparent lightpath from source to destination, or more likely a semi-lightpath consisting of several wavelengths; four steps are taken:

1. For a wavelength that is used for transparent switching at switch S, all arcs of the same wavelength coming in or going out of that switch will have their capacity set to 0. This means that one cannot separate part of the flow of a transparently switched wavelength at intermediate nodes.
2. The number of arcs that are used for transparent switching in the (semi-) lightpath will be collapsed into one arc in the capacity matrix, of the same capacity as the smallest arc. The cost of the new arc will be extracted from the function described in [15] and put in the cost matrix.
3. The capacity of the arcs on the shortest path is updated by subtracting from the arc capacities the new capacity granted.
4. The costs of all arcs on the (semi-) lightpath will be updated according to the cost function.

The same steps apply in the case of a low-priority bandwidth request. The difference lies in the use of the assigned low-priority cost matrix. We should note again that the identification of low-priority and high-priority flows is either done through Service Level Agreements or measurements. Measurements allow the algorithm to identify packets that

are traversing the switch coming from the same source, going to the same destination and requiring a large amount of bandwidth. A flow of such nature can be remarked from low-priority to high-priority flow to relieve the electronic switching load and make better utilization of the network resources.

With respect to the requests used in the simulation, the sources and destinations are random, allowing for wavelengths as well as packets from the electronic interfaces to come in to the switch, depending on whether the switch is acting as an ingress node or a core node. The values of bandwidths requested are random numbers distributed uniformly with values between 0 and 0.3 (30% of absolute link capacity). This is due to the fact that each wavelength's theoretical limit by today's technology is 10 Gbps. We assumed an upper limit of 3 Gbps reservation is acceptable. The criteria used to stop the simulation run is when the simulation receives several requests in a row that it cannot fulfill. An indeterminate number of requests are used until the algorithm has to reject 10 requests in a row. After the simulation runs, we consider results of accepted calls (requests), distribution of these calls, and the switching –whether optical or electronic– that they use.

5.8 Simulation Results

The simulation was developed using MATLAB. The shortest path algorithm is Dijkstra's algorithm. We used an implementation of this algorithm for sparse matrices developed by Mark Steyvers at Stanford University [48], which returns the length of the shortest path. We extended this code to also form an array representing the shortest path as well for further manipulation. We ran the simulation 20 times, with random sources, destinations, priorities and bandwidth needs changing every run. Then we found averages of all the information we were looking for. This information concerns the

utilization of the electronic vs. the optical switch, the ratio of low-priority flows that were switched electronically to the total number of low-priority flows, and the ratio of high-priority flows that were switched all-optically to the total number of high-priority flows.

For the topology presented earlier, the matrices of interest to us (namely the two cost matrices and the capacity matrix) showed the transformed graph with lightpaths that were setup, residual bandwidths and updated link costs. The transformed graph was a 75x75 matrix in order to account for the large number of wavelengths available. Table 5-3 shows the results that we got from the simulation.

Table 5-3 Results of The Simulation of the Heuristic Algorithm

Description	Value
Average utilization of wavelengths used in IP router in hybrid switch	92%
Average utilization of wavelengths connected to optical switches	72%
Number of flows accepted	185
Number of high-priority flows accepted	131
Total number of rejected requests	68
Total number of rejected low-priority requests	40
Number of high-priority flows that used IP routers	110
Number of high-priority flows that only	21

used optical switching	
Numbers of low-priority flows that used some optical switching	2

The average utilization of the wavelengths that were electronically switched reached more than 90%. This is to be expected, since the grooming capability of the electronic switches is virtually unlimited. The delays incurred in such a situation at the input buffers to the IP routers may not be acceptable for individual flows. Therefore, integrating more Diffserv functionality as described in Chapter 2 is beneficial. High-priority flows did not go through the optical cross-connects exclusively since we only form a limited number of lightpaths. Therefore, when crossing through the network, high-priority flows will go through some optical switching and some electronic switching. In case the flow is going through the IP router, it would be beneficial to have the scheduling algorithm of that IP router give better service to the high-priority flow, therefore increasing the Quality of Service (QoS) for that flow and decreasing the delay incurred in the switch.

The results also show several wavelengths remaining unutilized. This is due to blocking in the optical switching fabric. Since there is no possibility of utilizing some of the wavelengths, when a switch has an odd number (2M + 1) of neighboring switches, at least one wavelength will not be utilized between that switch and one other neighboring switch. To prove this, consider the following lemma.

Lemma

For a blocking optical switching that switches N wavelengths $\lambda_1, \lambda_2 \ldots \lambda_n$ per fiber with no wavelength conversion, if that switch is connected to an odd number 2M + 1 of fibers,

then the maximum number of wavelengths over all the ports that can be utilized is 2NM, leading to a maximum utilization of 2M/(2M+1).

Proof

$$
\begin{aligned}
&\exists \Lambda = 2M + 1 \\
&\exists P = N * \Lambda = N(2M + 1) \\
&\forall F_i, \forall F_{j \neq i}, if\ \lambda_{F_i} \Leftrightarrow \lambda_{F_j} \rightarrow matching_{\lambda} = matching_{\lambda} + 2 \\
&\sum\nolimits_{F} matching_{\lambda} = 2M \\
&\max \sum\nolimits_{F,\lambda} use_{\lambda} = N * 2M = 2MN \\
&\max Utilization = \frac{2MN}{P} = \frac{2MN}{N(2M+1)} = \frac{2M}{2M+1}
\end{aligned}
\tag{5.1}
$$

Let the optical switch have P = N (2M+1) ports. There are • wavelengths per fiber. For all fibers, if a wavelength of fiber i is matched to an output port of fiber j, the number of matchings increases by 2, decreasing the available number of ports (total wavelengths) by 2. The total number of matchings that can be done is therefore 2M. The maximum number of wavelengths used over all fibers is 2MN. This leads to a maximum wavelength utilization of 2M/(2M+1) ♦

The low-priority flows were switched electronically on the path between source and destination. This was an objective of the heuristic algorithm and it was met in the simulation runs. This is due to the high cost of optical switching for the low-priority flows. Some low-priority flows did use some lightpaths that were already setup, when no other path could be found that could sustain the bandwidth demand. We added a cut-off at which the cost of the path would be too high. This ensured that low-priority flows would not set up several lightpaths and still be switched optically. This keeps

wavelengths geared for optical switching for high-priority traffic. Choosing between high- and low-priority flows was uniformly distributed, as were the bandwidth demands, so low-priority flows were rejected more often due to the inability to set up more lightpaths and their very high cost for setting up. This led to 60% of all rejected requests to be low-priority.

Most (85%) high-priority flows used IP routers for part of their switching to the destination sought. This is due to the small number of lightpaths that can be setup before electronic switching cores have to be used. This suggests that re-optimization of the lightpaths that are set up would be beneficial to the overall network load distribution and our objective of pushing as many high-priority flows to use optical switching as much as possible.

5.8.1 Interpretation of Results

The non-utilization of some of the wavelengths due to blocking and the utilization of some electronic switching by high-priority flows leads to the conclusion that the optical wavelength conversion has an advantage over a network where this is not possible. This would allow the setting up of a transparent lightpath that would consist of two or more wavelengths. Optical wavelength conversion has become feasible in the past few months with the advances in optics and more specifically tunable lasers and fiber gratings. Their integration in switches is a matter of time and economics. Optical wavelength converters could be integrated in our hybrid switch model as a component of the switch, and could be modeled by adding the cost of utilization of that converter. This would assure that Dijkstra's algorithm would pick a path that has the least electronic switching and as few wavelength converters as possible.

Our hybrid switch model does not allow for dynamic allocation of wavelengths to

the optical or electronic switching cores. Some wavelengths are statically reserved for optical switching while some others are reserved for electronic switching. A better but much more expensive approach would consist of dynamically allocating the wavelengths to the best switching core. The dynamic allocation can be done to decrease the utilization of one of the switches or to be better able to setup a lightpath. We chose not to make use of that capability because it would make the design of the switch much more complex, where the number of Add-Drop Multiplexers would become equal to the number of wavelengths, since any wavelength could be terminated in such a model. Though the algorithm would perform more efficiently, given that flexibility, we chose not to make use of such a model until some tunable filter technology makes the process possible and economical.

In our simulation run, we also incurred a high utilization of the electronic switching cores to accommodate optical traffic that needed to be terminated and regenerated again. This was due to the restriction on the number of channels allowed per fiber, which caused the algorithm to find semi-lightpaths that included IP routers. This led to the use of some of the optical wavelengths that were set for low-priority flows, for high-priority ones. In this situation, it would be crucial to be able to differentiate between different priority flows within the IP Router to deliver better service to the high-priority flows. This would ensure that the network operator delivers the QoS level requested in the Service Level Agreement.

5.9 Conclusion

We have developed a simulation to assess the applicability of the heuristic algorithm and its behavior. That simulation was developed using MATLAB. It showed the good utilization of the network as well as heavy utilization of the electronic switching cores.

Lightpaths that were setup optically were less utilized, and high-priority traffic made use of both electronic and optical switching cores. The simulation showed the need for optical wavelength conversion since high-priority traffic was making use of electronic wavelength conversion to connect between two lightpaths that were already setup, because of the inability to find one wavelength for the entire path. The simulation also showed the need for dynamic wavelength assignment inside the switch to either the optical or electronic switch core to increase the flexibility in the assignments. The results of the simulation allow us to indicate empirically that the heuristic algorithm produces results in accordance with the objectives that were set out in terms of network loading and load balancing and the overall performance of the switch and the associated algorithm.

Chapter 6

Conclusion

We have proposed a hybrid switching system that takes advantage of both packet switching and circuit switching to deliver better network performance. The resulting performance improvement stems from the ability to make intelligent decisions on the kind of switch that will make best use of the network resources. The Internet infrastructure is changing fast. The need for more bandwidth, the new services required by customers and the general advances in fiber optic transmission lines have created a combination of factors that require more sophisticated telecommunications equipment.

6.1 Summary of Work

The thesis discussed an architecture that supports Differentiated Services, a protocol that allows the network to prioritize traffic based on Service Level Agreements. Multi-Protocol Label Switching allowed us to pin the routes assigned to the different traffic

flows. The architecture provided traffic streams with varying combinations of bounds on QoS metrics such as minimum bandwidth, maximum delay and maximum jitter. We also described different restoration methods that could be used in the case of link failures in such a traffic-engineered environment.

This work emphasized the importance of being able to deal with link failures. In an MPLS environment such as the one described earlier, restoration can be done relatively easily. This is not the case of link-state routing where different switching nodes see different shortest paths depending on what nodes have sent them Link State Advertisements (LSA). The first part of the thesis addressed that issue. In that part, we considered the extensions to link-state routing protocols such as Open Shortest Path First (OSPF). Those extensions allow OSPF, a time-tested protocol suite, to support traffic engineering. This is achieved through algorithms that make judicious cost assignments to the individual links. Link failures make those cost assignments sub-optimal. Running the cost-setting algorithm allows for a re-optimization of the network load, but as was proven, leads to severe routing problems and routing loops. Therefore, we devised an architecture that allows better support for link failures. This architecture is based on notification of selected nodes in the network to update their information and their costs, and to make routing decisions based on their view of the network. Informed nodes can assess the ability to find loop-free routes for all downstream routers whether informed and uninformed of the link failure. The algorithm works by constructing first a restoration path around the link failure and informing the nodes on that path of the failure. It then increases the number of informed nodes by informing neighboring nodes of that failure and making them part of a restoration network that grows to include all the nodes in the network. We discussed the routing decisions made by every node as well as

a proof the correctness of the algorithm. This algorithm deals with single link failures and its applicability has not been verified with multiple link failures.

The second part of the thesis is motivated by the ability to provide through OSPF extensions the restoration capability required in QoS networks. After looking at the current state of the switching technology, it became apparent that scalability and services were of utmost importance. With the explosive growth of IP networks and increased reliance on the network infrastructure for business needs, there is need to increase bandwidth, service availability and reliability. We built a framework for a new switch design that combines packet switching and circuit switching through the use of an IP router and an optical cross-connect. The approach is premised on the need for servicing differing traffic needs and differing customer sets. Traffic can be classified in two broad categories of high-priority and low-priority flows. This allowed us to scale through the optical switching capability for the high-priority flows while providing bandwidth granularity through electronic switching to the low-priority flows.

With respect to the static flow demand, which includes requests that are made through SLA's, we built an Integer Programming formulation that optimizes the network resource utilization by distributing the load over several switching nodes, and within each switching node, through both the optical and electronic switching cores. The objectives of keeping bounds on the utilization of the OXC and the IP router and link resources were met through the Integer Program.

The Integer Programming formulation allowed for an optimal distribution of the network load based on the objective function and the constraints. It did not address the need for requests coming dynamically to the network edge to find paths to the destinations. We devised an algorithm to deal with this shortcoming that is based on

heuristics and does not assume *a priori* knowledge of future demand. The heuristic algorithm makes its best effort to route the traffic through optical cross-connects and electronic routing cores by making intelligent cost assignments to the links in the network. To accommodate for both low-priority and high-priority flows, the algorithm takes a multi-commodity flow approach by assigning higher costs for links geared for high-priority traffic to any low-priority traffic that would use them and *vice versa.* Simulation results over a sample network topology were presented that identified both the current algorithm behavior and areas for future improvement.

6.2 Improvements and Future Work

The simulation results as well as the research that were conducted identified several areas where more research would be of interest and value. One key area is optical wavelength conversion. This would allow the algorithm to find lightpaths more easily since it is not restricted to one wavelength per lightpath.

The work presented in this thesis addresses the routing assignments based on a certain number of wavelengths being geared to optical switching while the others are allocated for the electronic routing. The restriction on which wavelengths to send to the optical cross-connect and vice versa is an engineering decision based on cost issues. Since the purpose of the engineering of the hybrid switch is to build a cost-effective solution, it is important to keep the number of components small. In today's technology, were we to supply an Add Drop Multiplexing component at each wavelength, the number of ADM components needed would be directly related to the number of wavelengths in total. Scalability issues would be again a concern. Developments in component design and manufacturing have been taking place especially in the area of tunable filters to keep the number of components smaller, where a proportion of the wavelengths would be

geared to electronic switching, therefore only requiring the same amount of ADM components. With that technology becoming readily available, enhancements to the heuristic algorithm would take into account that added capability for better performance and path assignments.

Bandwidth protection through lightpath protection or fiber protection is a very important issue that needs to be addressed to allow the deployment of such technology. Protection refers to the reservation of extra bandwidth on a different path than the path used for the transmission. In case the transmission path incurred a failure, the network would be able to re-route its traffic on the protection path almost instantaneously. This is a requirement from the telecommunications carriers, especially in the area of voice traffic. SONET technology, used primarily in voice networks, guarantees a maximum failover time between primary and protection path of less than fifty milliseconds. While this is not a requirement of data networks, heterogeneous environments such as the ones discussed in this thesis would have to make failover time guarantees to high-priority flows, which could well include voice traffic. The protection problem is especially important in the case of fiber protection, since a fiber cut would lead to the disruption of several lightpaths, which, at the lightpath layer, would look like multiple link failures. Work in this area should address the multiple lightpath failures that occur in this case.

Finally, it would be beneficial to have the heuristic algorithm aware of the traffic that it has accepted in the network. This would allow it to build a new routing and wavelength allocation based on the traffic that has already been accepted, leading to more optimal traffic routing. In addition, more intelligence could be built in learning about traffic request patterns. If the algorithm sees a lot of low-priority flow requests coming at a certain node, it could make inferences about the future traffic flows to be able to better

service them.

This thesis builds a framework for hybrid packet and circuit switching that we hope will be the basis for future networks and sheds new light on the problems that those networks will face and solve. The author hopes that the thesis helps in identifying numerous research topics of use to the networked world in general and the networking community in particular.

References

[1] Blake, S., D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," Request For Comment, RFC 2475, December 1998.

[2] Heinanen, J., F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," Request For Comment, RFC 2597, June 1999.

[3] Jacobson, V., K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," Request For Comment, RFC 2598, June 1999.

[4] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", Request For Comment, RFC 792, September 1981.

[5] Moy, J., *OSPF: Anatomy of an Internet Routing Protocol*, Reading: MA, Addison-Wesley Publishing Company, 1998.

[6] Awduche, D. O., J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering over MPLS," Request For Comment, RFC 2702, September 1999.

[7] Apostolopoulos, Dimitrios, R. Guerin, S. Kamat, Orda, T. Przygienda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions", Internet Draft, April 1998.

[8] Zhang, Z., C. Sanchez, B. Salkewicz and E. Crawley, "Quality of Service Extensions to OSPF or Quality of Service Path First Routing", Internet Draft, September 1997.

[9] Braden, R., L. Zhang, S. Berson, S., Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[10] Ma, Q. and P. Steenkiste, "Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks," Workshop on Network and Operating Systems Support for Digital Audio and Video, Cambridge, England, July 1998.

[11] Parekh, A., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks, Ph.D. Dissertation, MIT, Cambridge, MA, February 1992.

[12] Ahuja, Ravindra, Thomas Magnanti, and Jim Orlin, *Network Flows: Theory, Algorithms, and Applications.* Upper Saddle River, NJ: Prentice-Hall, Inc., 1993.

[13] Li, T. and Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)," Request For Comment, RFC 2430, October 1998.

[14] Moore, Geoffrey A., Regis McKenna, *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*, HarperCollins, 1991.

[15] Fortz, Bernard and Mikkel Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights", Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[16] Rosen, Eric, A. Viswanathan and R. Callon, "Multi-Protocol Label Switching Architecture," Request For Comment, RFC 3031, July 2001.

[17] Narvaez, Paolo, K. –Y. Siu, H. – Y. Tzeng, "Local Restoration Algorithm for Link-State Routing Protocols", Proceedings of the 1999 IEEE ICCCN, Natick, Massachusetts, October 1999.

[18] Labovitz, C., G. Malan and F. Jahanian, "Internet Routing Instability", Proceedings of SIGCOMM '97, Cannes, France, September 97.

[19] Perlman, R. and G. Varghese, "Pitfalls in the design of distributed routing algorithms", Proceedings of SIGCOMM '98, Vancouver, Canada, August 1998.

[20] Bertsekas, Dimitri and Robert Gallager, Data Networks, Prentice-Hall, Inc., New Jersey: 1987.

[21] Shaikh, Anees, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows", in Proceedings of SIGCOMM, September 1999.

[22] Newman, P., G. Minshall and T. Lyon, "IP Switching: ATM under IP," IEEE/ACM Transactions on Networking, Vol. 6, pp. 117-129, April 1998.

[23] Banerjee, Dhritiman and Biswanath Mukherjee, "Wavelength-Routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs, and a Reconfiguration Study", in IEEE/ACM Transactions on Networking, Vol. 8, No. 5, October 2000.

[24] Chlamtac, Imrich, A. Faragó and T. Zhang, "Lightpath (Wavelength) Routing in Large WDM Networks", IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, June 1996.

[25] "Optical Switching Fabric", Report, Light Reading, October 30, 2000, report available online at Light Reading's website at URL http://www.lightreading.com/document.asp?doc_id=2254.

[26] Cinkler, Tibor, D. Marx, C. P. Larsen, and D. Fogaras, "Heuristic Algorithms for Joint Configuration of the Optical and Electrical Layer in Multi-Hop Wavelength Routing Networks", Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[27] Iwata, Atsushi, Norihito Fujita, Gerald Ash and Adrian Farrel, "Crankback Routing Extensions for MPLS Signaling with RSVP-TE", Internet Draft, Work in Progress, November 2000.

[28] Narvaez, Paolo, K. –Y. Siu and H. –Y. Tzeng, "New Dynamic SPT Algorithms based on a Ball-and-String Model", Infocom 99, Proceedings of the IEEE Conference on Computer Communications, Vol. 2, New York, New York, March 1999.

[29] Kodialam, Murali and T. V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks", Proceedings of IEEE INFOCOM 2001, April 2001.

[30] Awduche, Daniel O., A. Chiu, A. Elwalid, I. Widjaja, X. Xiao, "A Framework for Internet Traffic Engineering", Internet Draft, Work in progress, April 2001.

[31] Dixit, Sudhir and Yinghua Ye, "Streamlining the Internet-Fiber Connection", IEEE Spectrum, pp. 52-57, April 2001.

[32] Gibran, Khalil, *The Prophet.*

[33] Comer, Douglas E., *Internetworking with TCP/IP volume 1 Principles, Protocols and Architecture*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995.

[34] Bertsekas, Dimitri and Robert Gallager, *Data Networks Second Edition*, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1992.

[35] Stallings, William, *High-Speed Networks, TCP/IP and ATM Design Principles*, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1998.

[36] Peterson, Larry, *Computer Networks: A Systems Approach*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.

[37] Lucent Technologies, Inc., "Bell Labs Researchers Use 'Ultra-Dense WDM' to Transmit Data Over 1,022 Channels on a Single Optical Fiber", Murray Hill, New Jersey, November 10, 1999, press release available online at Lucent Inc., at URL http://www.lucent.com/press/1199/991110.bla.html.

[38] Rabbat, Richard, Ken Laberteaux, Nirav Modi and John Kenney, "Traffic Engineering Algorithms Using MPLS for Service Differentiation", Special Session on Internetworking of Diffserv, RSVP and MPLS for achieving QoS in the Internet, *Proceedings of the International Conference on Communications*, ICC 2000, New Orleans, June 2000

[39] Rabbat, Richard and Kai–Yeung Siu, "Restoration Methods for Traffic Engineered Networks with Loop-Free Routing Guarantees", *Proceedings of the International Conference on Communications*, ICC 2001, Helsinki, June 2001.

[40] MCI WorldCom, Investor Relations, Building the Right Networks, Annual Report, 1997, available online at MCI-WorldCom, Inc at URL http://www.worldcom.com/investor_relations/annual_reports/1997/networks.

[41] Level 3 Communications, Building the Network, Network Plan (plan), available online at Level 3 Communications, Inc.'s website at http://www.level3.com/us/info/network/network_map.

[42] Rajagopalan, Bala, Dimitrios Pendarakis, Debanjan Saha, Ramu S. Ramamoorthy and Krishna Bala, "IP over Optical Networks: Architectural Aspects", IEEE Communications Magazine, vol. 38, no. 9, pp. 94-102, September 2000.

[43] Listanti, Marco, Vincenzo Eramo and Robert Sabella, "Architectural and Technological Issues for Future Optical Internet Networks", IEEE Communications Magazine, vol. 38, no. 9, pp. 82-92, September 2000.

[44] Leland, W. E., *et al*, "On the Self-Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking, vol. 2, no. 1, pp. 1-15, 1994.

[45] "LuxN: New Customer, New Product", Report, Light Reading, April 27, 2001, report available online at Light Reading's website at URL http://www.lightreading.com/document.asp?doc_id=4922.

[46] "Pirelli and AT&T Labs Announce Multi-wavelength 40 Gbps Breakthrough on Single-Mode Fiber", Press Release, AT&T, May 21, 1999, report available online at AT&T's website at URL http://www.att.com/technology/press/990521.html.

[47] The MathWorks, MATLAB, simulation and technical computing software, 2001.

[48] Kay, Michael G. and Mark Steyvers, Dijkstra's algorithm implementation for MATLAB simulation, software, December 19, 2000.