PROTECTION ALGORITHMS FOR BANDWIDTH GUARANTEED CONNECTIONS IN MPLS NETWORKS

WONG SHEK YOON

(B.Eng.(Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgments

The author would like to express her gratitude to her supervisor Dr. Mohan Gurusamy for his continued guidance and invaluable advice that has made this research possible.

Contents

Summary

Network survivability has become increasingly important for emerging applications (time sensitive or mission critical applications, e.g. multimedia services, video conferencing, telemedicine applications, etc.) in the rapidly growing Internet. This dissertation focuses on presenting path protection algorithms in multiprotocol label switching (MPLS) networks.

MPLS technology is a key technology for the implementation of scalable virtual private networks (VPNs) and providing end-to-end quality of service (QoS). It enables efficient utilization of network resources to meet future growth and rapid fault recovery of link (node) failure. In case of failure, MPLS uses a new label switched path (LSP) and then forward the packets to the new LSP.

In this dissertation, multiple backup paths are used to protect primary connection. Multipath routing is used to route backup paths and to reduce the load on the congested links in a network. The amount of bandwidth wasted is reduced when multiple backup paths are used, and all the backup bandwidth can be shared by backup connections whose primary paths are link disjoint. Two efficient multipath protection algorithms are presented in this dissertation. The equal splitting multipath protection algorithm distributes load equally on every backup paths. On the other hand, the failure dependent multipath protection algorithm tries to allocate different amount of load to every backup paths for every primary path link of a demand.

To improve the routing performance of dynamic traffic in distributed MPLS networks, additional information can be disseminated among network routers. Three supplementary information are introduced to further improve the performance of demands routed. An algorithm using only shortest path computation is developed to route connections with the supplementary information proposed. Through extensive simulations performed, it is found that these supplementary information help to reduce the number of request dropped, as well as balance the traffic in the network.

In the next part of the thesis, scheduled demands are considered where, in addition to the source and destination of a LSP connection, the setup time and teardown time are known in advance. This allows bandwidth resources in MPLS networks to be reused by scheduled LSP demands that are not time overlapped. The routing of scheduled LSP demands in a MPLS network can be solved using integer linear programming (ILP) to obtain optimal solutions. However, ILP is not scalable and thus not desirable for routing of demands in large networks. A fast heuristic algorithm has been proposed. This algorithm identifies groups of LSP demands that are time overlapped with each other and then solves each group using a fast and efficient algorithm. Experiments show that the algorithm proposed has performance close to that of the ILP.

List of Figures

List of Tables

List of Acronyms

CR-LDP	Constraint-based Routed Label Distribution Protocol
CSPF	Constrained Shortest Path First
DiR	Differentiated Reliability
DPIM	Distributed Partial Information Management
FECs	Forwarding Equivalence Classes
ILM	Incoming Label Map
ILP	Integer Linear Programming
LDM	Load Distribution over Multipath
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switching Router
MATE	Multipath Adaptive Traffic Engineering
MHA	Min-Hop Algorithm
MHD	Multi-Hour Design
MIP	Mixed Integer Programming
MPLS	Multiprotocol Label Switching

NHLFE	Next Hop Label Forwarding Entry
OSPF	Open Shortest Path First
PROMISE	PROtection using Multiple Segments
PSI	Protection with Supplementary Information
QoS	Quality of Service
RSVP-TE	Resource Reservation Protocol Traffic Extension
RWA	Routing and Wavelength Assignment
SCI	Sharing with Complete Information
SHD	Single-Hour Design
SLA	Service Level Agreement
SPF	Shortest Path First
VPN	Virtual Private Network
WDM	Wavelength Division Multiplexing
WSP	Widest Shortest Path

CHAPTER 1

Introduction

The demand for high transmission capacity is growing at an unprecedented rate driven by the rapidly growing Internet. Ubiquitous and frequent congestion situations have restricted the use of new time-critical applications, e.g. IP telephony, video conferencing and online games. Thus, there is not only an increasing demand for bandwidth but also some sort of scalable quality of service (QoS) support. Besides that, the failure of a link or router may have severe effects on these applications. The interruption caused by a link or router failure may be too long for real time services to maintain their sessions and certain QoS (e.g. delay, jitter) may not be achieved due to the failure. Therefore, it is important that the issue of network resilience be studied to guarantee the performance of the Next Generation Networks.

1.1 Driving Forces for The Protection Algorithms Proposed

The main driving forces of proposing the protection algorithms in this dissertation come from the need of providing fast and efficient network failure restoration. As the telecommunication market is evolving towards services, network management becomes increasingly complex. Network survivability becomes one of the major issues in Service Level Agreements (SLA) as applications in the Next Generation Networks require high bandwidth connections that are reliable and at the same time optimize bandwidth resources utilization. Multiprotocol label switching (MPLS), which is a path-oriented forwarding technology, is a good solution to support survivability requirements and to enhance the reliability of networks in providing services.

Fast algorithms, which utilize MPLS network resources efficiently and route reliable connections, are crucial to provide satisfying network services to the ever-increasing Internet's user population. There are basically two ways to restore traffic after a network link or router fails: protection and restoration. In protection scheme, additional backup resources are reserved when a connection is set up. This allows affected traffic to be re-routed rapidly after network failure. On the other hand, restoration scheme tries searching resources for failure-affected traffic only after network component failure. As a result, no guarantee can be given to provide reliable transmission in network using restoration scheme. This characteristic of restoration making it unsuitable for applications whose QoS is a major consideration.

Even though protection scheme, which provides fast restoration, guarantees the restoration of failure-affected traffic in network, it contradicts the objective to utilize network resources efficiently. It is a challenging problem to optimize network resources for more network users while at the same time, provide reliable connections with QoS requirements fulfilled to applications require high performance. It is the objective of this dissertation to optimize the performance of MPLS network with guaranteed protection by using effective routing, and additional information of traffic statistics, traffic characteristics and network topology.

1.2 Related Work

In [1], [2], [3], the basic architectures and concepts of MPLS are given. Two main methods are proposed for network survivability in MPLS networks: protection and restoration. In this dissertation, protection method is used to provide fast and reliable traffic restoration.

Research works in [4], [5], [6], [7], [8], [9] give ideas of providing networks survivability using protection method. In [4], [5], [6], [7], [8], path protection is given through the reservation of spare bandwidth on backup paths. In case of failure, the failure information must be propagated to the source node. The source node of each affected connection is responsible for switching the affected traffic to the spare backup path reserved. Link state information is disseminated and used to determine the amount sharing that can be achieved between link-disjoint backup paths. Through backup bandwidth sharing, the amount of spare backup bandwidth consumed in the network can be reduced while providing bandwidth guaranteed protection.

In [9], a link protection scheme for dynamic traffic was proposed. Link protection means that upon a link or node failure, the first node upstream from the failure must be able to switch the path to an alternate preset outgoing link so that the traffic affected is restored by a local decision. A bypass backup path for every link and node used by the primary path must be set up together with the primary path when a label switched path (LSP) setup request is accepted. Link protection is much faster than path protection because failure information does not have to propagate to the source.

As traffic engineering and quality of service (QoS) become important to provide better network efficiency and transmission quality (e.g. delay, blocking probability), these issues were taken into consideration in research works [10], [11], [12]. End-to-end transmission delay from the source to the destination is an important QoS constraint especially for multimedia applications like video communications. The switch-over time, which is the time for which the packets will be dropped over the primary LSP after a failure, is also an important QoS constraint determining how fast a failure can be restored. Traffic engineering techniques aim to optimize the utilization of network resources for failure recovery.

To further improve network throughput, multipath routing is adopted in [13], [14], [15], [16], [17], [18], [19] and [20]. In multipath routing, traffic with the same source and destination are allowed to take more than one possible path in order to relax the most heavily congested link. Two issues have to be considered in a multipath routing algorithm: computation of multiple loop-free paths and traffic splitting among these multiple paths. The traffic can be split based on specified or derived load ratios. The ratio may be proportional to the lengths of paths or to the bandwidths of the label switched paths (LSPs). Multipath routing helps to ensure more even distribution of load in MPLS networks.

A kind of traffic called scheduled demands was introduced in [21] and [22]. A scheduled demand is a connection demand whose setup and teardown times are known in advance. In [21], the routing problem and wavelength assignment problem were separately formulated as spatio-temporal combinatorial optimization problems. A branch and bound algorithm was proposed to compute exact solution and a tabu search algorithm was proposed to compute approximate solution to the optimization problems. Integer linear programming formulations of scheduled demands was developed for both dedicated and shared path protection WDM-based networks in [22]. The time disjointness that exists among scheduled demands is captured to reduce the total amount of network resources required.

1.3 Algorithms Proposed and Their Novelties

In this dissertation, all the algorithms proposed for MPLS networks were developed with the following characteristics:

- Provide reliable LSP connections - Path protection is achieved in every algorithm presented. For every LSP connection that is protected using path protection scheme, a backup LSP has to be established at the point of primary LSP setup. In case of a failure on the primary LSP, the source label switching router (LSR) on receiving the failure notification can reroute the affected traffic over the backup path. This allows traffic to be rerouted immediately once failure notification is received and traffic restoration is guaranteed with backup bandwidth pre-reserved.

- Fast computations - Algorithms proposed are fast heuristics that do not require complicated calculations. This allows routing decisions to be made without consuming large amount of computing resources.

- Resource efficient algorithms - Algorithms developed aim to utilize network resources efficiently so that more requests can be satisfied. Shortest path routing, multipath routing and backup bandwidth sharing are techniques used to reduce redundancy incurred.

- Scalability - Fast and efficient algorithms provide scalability to service LSP demands in large networks.

1.3.1 Multipath Protection

Instead of using single backup path, multiple backup paths are used to protect the primary path of a demand. In the multipath protection algorithms proposed, bandwidth guaranteed backup paths are established when the primary path is set up. This ensures sufficient bandwidth to route affected traffic upon network failure. Multipath routing helps to reduce congestion links in a network. This is because multipath routing can evenly distribute the traffic load over the network. The performance of multipath routing can be further improved when multipath routing is used to route backup paths. Backup bandwidth sharing is adopted to reduce the redundant spare bandwidth caused by multipath routing.

Two multiple backup path routing algorithms are presented in this dissertation. The equal splitting multipath protection algorithm distributes traffic load of a demand equally on every backup paths. The second algorithm, which is called failure dependent multipath protection, does failure dependent load balancing. That is, for every link on primary path, different proportion of traffic load will be allocated to each backup path. The simulation results show that both the algorithms proposed perform better than single shortest path algorithm.

1.3.2 Protection with Supplementary Information

In distributed networks, link state information is important for making routing decisions. Three basic link information required to allow backup bandwidth sharing includes aggregate bandwidth used by primary connections, aggregate bandwidth used by backup connections, and free residual bandwidth available on a link. These three basic link information is called partial information in [4].

In this dissertation, additional link state information is exploited to improve performance of reliable networks. Supplementary information proposed includes primary bandwidth change, backup bandwidth change, and distribution of traffic in the network for every source and destination pair. The average connection holding time and average arrival rate of demands need to be maintained at every node to derive the primary bandwidth change and the backup bandwidth change on links connected to the node. The information of source and destination pairs' traffic distribution is local to every ingress label edge router (LER). That is, every ingress LER just need to know the traffic distribution of source and destination pairs that go through itself.

The supplementary link state information proposed helps to balance the traffic distribution in a network as well as improve backup bandwidth sharing, so that more LSP setup requests can be accepted. A fast and efficient heuristic algorithm using the supplementary information is described in this dissertation. Extensive simulations have been conducted to verify the effectiveness of the proposed algorithm in different scenarios. It is found that the algorithm developed with supplementary information provides efficient resource utilization and better network performance than algorithms using only partial information.

1.3.3 Survivability of Scheduled LSP Demands

Scheduled LSP demands with its setup time and teardown time known are studied. The performance of scheduled LSP demands in MPLS networks is good as network resources can be reused by scheduled LSP demands that are time disjoint with each other. Besides that, by knowing the setup time and teardown time of demands, efficient primary and backup paths routing can be achieved.

Characteristics of scheduled LSP demand are analyzed to present fast algorithm that produces routing solutions very close to the optimal ones obtained from integer linear programming (ILP). In the fast algorithm proposed, first, scheduled LSP demands in the MPLS network are sorted into groups. Every demand in the same group is time overlapped with each other. Then, scheduled LSP demands in the same group are routed using the heuristic proposed. Experiment results show that the heuristic proposed for scheduled LSP demands improves network resource utilization and reduces the number of demands dropped.

1.4 Organization

In Chapter 2, the basic concepts of multiprotocol label switching (MPLS) is explained in detail. Chapter 3 reviews existing network survivability in MPLS networks. In Chapter 4, a protection scheme using multiple backup paths is discussed. Chapter 5 elaborates improved reliable routing with the consideration of the new traffic information. Chapter 6 presents the survivability of scheduled LSP demands in MPLS networks. Extensive simulations are performed to evaluate the performance of all the algorithms proposed in this dissertation. Finally, Chapter 7 gives a summary of the contributions and concludes the dissertation.

CHAPTER 2

Multi Protocol Label Switching

2.1 Introduction to MPLS

As many emerging applications (e.g. time-critical applications: video conferencing, remote medical diagnosis) requiring high transmission capacity and reliable bandwidth guaranteed connections, the need to provide fast restorable bandwidth guaranteed paths becomes an important issue in the next generation networks. Multiprotocol label switching (MPLS) technology, which enables the service providers to traffic engineer the networks, is considered a favorable solution that addresses the needs of future IP-based networks.

MPLS protocol is introduced by the Internet Engineering Task Force in [1]. MPLS is a set of open, standards-based Internet technologies that combines layer 3 routing (IP) with layer 2 switching technologies (e.g. ATM) to forward packets by utilizing short, fixed-length labels. Therefore, MPLS is capable to provide the best of both layers: the efficiency and simplicity of routing, as well as the high speed of switching. With suitable routing algorithms, network resources can be utilized efficiently in MPLS networks.

Multiprotocol label switching is a connection-oriented protocol. That is, at every ingress router (ingress label edge router, LER) packets that has the same attributes (e.g.

destination, quality of service and etc.) are classified into forwarding equivalence classes (FECs). There is no further packet header analysis at subsequent core routers. All packets that belong to a particular FEC will follow the same path. The FEC to which the packet is assigned is encoded as a short fixed length value known as label. Labels are used to forward the packets along the label switched path (LSP). At core routers (label switching router, LSR), after a label is examined to find the next hop, a new label will be issued to replace the old label before the packet is forwarded to its next hop.

2.2 MPLS Basics

This section introduces the basic concepts of MPLS based on the definitions given in [1].

2.2.1 Control and Data Planes

MPLS operates with control plane and data plane. The control plane uses different protocols (e.g. RSVP-TE, LDP/CR-LDP, OSPF-TE and IS-IS-TE) to perform a variety of operations, including:

- Information dissemination: The control messages are exchanged between nodes to establish a relationship so that label/FEC binding information can be exchanged. Periodic messages are forwarded to make sure that neighbour nodes are up and running. Besides that, routing information between routers and label binding procedures for converting routing information into forwarding table are distributed. The link state information is disseminated through protocols in control plane and is crucial to the path selection, path establishment and maintenance functions. Both

OSPF and IS-IS protocols have been extended to include resource style information about all links in the specific area. Through these extensions, MPLS traffic engineering becomes possible.

- Path selection: The control plane determines the best path of a LSP through a network.

- Path establishment and maintenance: Once a path has been determined, a signaling protocol is used to establish the new LSP. The signaling protocol distributes the specifications of the path, including the session identification and resource reservations to all other routers in the path. After a LSP is established successfully, the control plane is responsible to maintain the established LSP.

The function of the data plane is to forward all data packets by examining the label in the MPLS packet header. First, packets that arrive at the ingress label edge router (LER) are classified into forwarding equivalence classes (FECs). Then the LER pushes the applicable labels on the packets. Label switching routers (LSRs) along the label switched path (LSP) will forward the labeled packets based on the top label in label stack. The label switched path terminates at the boundary between an MPLS enabled network and traditional network. Finally, the egress label edge router removes the label from a packet and forwards the packet based on the packet's original contents, using traditional means.

2.2.2 Forwarding Equivalence Class

A forward equivalence class (FEC) is a group of packets that can be handled (i.e., forwarded) in the same manner. Therefore, packets in the same FEC are forwarded over the same path with the same forwarding treatment and thus are suitable for binding to a single MPLS label at the ingress LER. FEC allows the grouping of packets into classes based on some attributes, e.g. destination, precedence, quality of service (QoS) information and etc.

2.2.3 Label

After a packet is examined at the ingress LER, a MPLS shim header or label is attached to the packet. This label is short and fixed length. The shim header is pushed between layers 2 and 3 for a IP packet (layers 2- 7 of OSI model) (see Figure 2·1). Shim header provides a mean to relate layer 2 and layer 3 information. There are 32 bits in a shim header, out of which 20 bits are used for the label, 3 bits for experimental (exp) functions, one bit for stack (stk) function, and 8 bits for time to live (TTL).

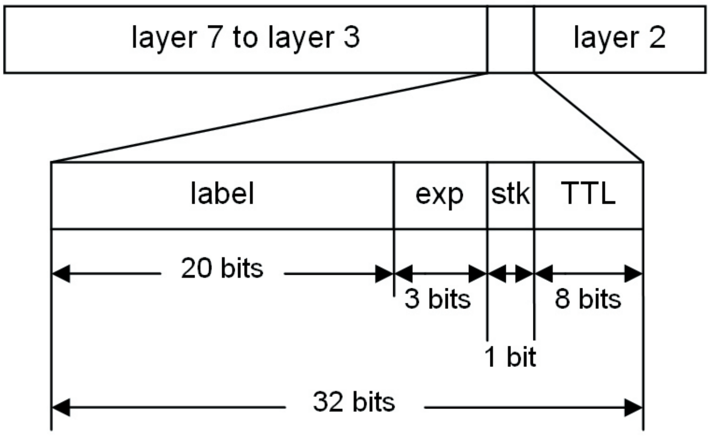

Figure 2·1: The MPLS Label and Format

2.2.4 The Label Stack

MPLS supports hierarchical operations as it allows more than one label in a packet. A labeled packet can carry a number of labels which is organized as a last in, first out stack. If a packet label stack is of depth *m*, the label at the top of the stack is called level *m* label, the label below the top label is called level $m - 1$ label, and the label at the bottom of the stack is level 1 label. Whenever an LSR pushes a label onto an already labeled packet, it needs to make sure that the new label corresponds to a FEC whose LSP egress router is the LSR that assigned the label which is now second in the stack. The processing of a labeled packet is independent of the level of hierarchy. This is because the processing is always based on the top label, without regard for other labels in the label stack.

2.2.5 Upstream and Downstream LSRs

If two LSRs *Ru* and *Rd* agree to bind label *L* to FEC *F* for packets sent from *Ru* to *Rd*, then with respect to this binding, *Ru* is the "upstream LSR" and *Rd* is the "downstream LSR".

2.2.6 Label Assignment

The decision to bind a particular label *L* to a particular FEC *F* is made by the downstream LSR *Rd* with respect to that binding. The downstream LSR *Rd* then informs the upstream LSR *Ru* of the binding. *L* is an arbitrary value whose binding to *F* is local to *Ru* and *Rd*. *Rd* must make sure that the binding from label to FEC is one-to-one. Label distribution protocol (LDP) is used to inform LSRs of the label/FEC binding made. Each

LSR must make sure that it can uniquely interpret its incoming labels.

2.2.7 Label Distribution Protocol (LDP)

A label distribution protocol is a set of procedures by which one LSR informs another of the label/FEC bindings it has made. Label distribution peers are two LSRs which use LDP to exchange label/FEC binding information. The label distribution peers are with respect to some set of binding information they exchange only, not with respect of some other set of bindings.

2.2.8 Label Switched Path (LSP)

An LSP of level *m* begins with an LSR (an "LSP Ingress") that pushes on a level *m* label. Thereafter, all intermediate LSRs make their forwarding decisions by label switching on a level *m* label. The LSP ends at an LSR (an "LSP Egress") when a forwarding decision is made by label switching on a level $m - k$ label, where $k > 0$, or when a forwarding decision is make by non-MPLS forwarding procedures. A sequence of LSRs is called the "LSP for a particular FEC *F*" if it is an LSP of level *m* for a particular packet *P* when *P*'s level *m* label is a label corresponding to FEC *F*.

The Label Distribution Protocol (LDP), Border Gateway Protocol (BGP) and Intermediate System to Intermediate System (IS-IS) protocols establish the label switch path (LSP), but do little for traffic engineering. To overcome this problem, the signaling protocols are used to create traffic tunnels (explicit routing) and allow for better traffic engineering. They are Constraint-based Routed Label Distribution Protocol (CR-LDP) and Resource Reservation Protocol (RSVP-TE). In addition, the Open Shortest

Path First (OSPF) routing protocol has undergone modifications to handle traffic engineering (OSPF-TE).

2.2.9 LSP Tunnels

The end-to-end LSP is called an LSP tunnel. The characteristics of the LSP tunnel (e.g. bandwidth allocation) are determined through negotiations between LSRs. The set of packets that are to be sent through the LSP tunnel constitutes a FEC. After the LSP is set up, packets are forwarded through the tunnel based on the label given; no further examination is made.

2.2.10 The Next Hop Label Forwarding Entry (NHLFE)

The Next Hop Label Forwarding Entry (NHLFE) is used when forwarding a labeled packet. It contains the following information:

1. the packet's next hop

2. the label value to be replaced

3. the label stack to be added to an MPLS-encoded packet

4. the data link encapsulation to use when transmitting the packet

5. the way to encode the label stack when transmitting the packet

2.2.11 Incoming Label Map (ILM)

When forwarding packets arrive as labeled packets, the Incoming Label Map (ILM) maps each incoming label to a set of NHLFEs. If the ILM maps a particular label to a set of NHLFEs that contains more than one element, exactly one element of the set must be chosen before the packet is forwarded.

2.2.12 FEC-to-NHLFE Map (FTN)

FTN correlates each FEC to a set of NHLFEs. It is used when forwarding packets arrive unlabeled at MPLS LER. These packets need to be labeled before being forwarded to next hop. If the FTN maps a particular label to a set of NHLFEs that contains more than one element, exactly one element of the set must be chosen before the packet is forwarded.

2.2.13 Label Swapping

The label does not retain the same value when a packet travels through the LSP. When a labeled packet arrives at a LSR, the label at the top of the label stack is examined. ILM is used to map this label to an NHLFE. Based on the information in the NHLFE, the LSR determines where to forward the packet, and changes the value of the label. In order to forward an unlabeled packet, first, the LER will analyze the network layer header to determine the packet's FEC. After that, FTN is used to map the packet's FEC to an NHLFE. The next hop is taken from the NHLFE. The LER then encodes the new label stack into the packet and forwards it.

2.3 Advantages of MPLS

This section briefly discusses several advantages of MPLS. MPLS enables traffic engineering. Explicit traffic routing and engineering in MPLS network help squeeze more data into available bandwidth. In addition to that, MPLS also supports the delivery of services with guaranteed quality of service (QoS). MPLS is not restricted to any specific link layer technology. It can be applied in many different network environments.

2.3.1 Quality of Service Support

For demanding types of applications (e.g., voice, multimedia), the best effort approach is not a very good model for transmitting data. It has become increasingly evident that Internet needs to differentiate between types of traffic and to treat each type differently. The service needs of different applications can be represented as a set of parameters, which include bandwidth, delay, jitter, packet loss, preemption and some others. Applications such as voice and multimedia are very sensitive to delay and jitter, whereas some data applications may require very low packet loss. In addition to that, when offering quality of service (QoS), an application flow traversing the network should receive the appropriate class-based treatment.

MPLS can support the quality of service mentioned above. Label switching is fast as only the label is used to index into the forwarding table at routers. As a result, MPLS reduces the delay and response time to enact a transaction between users. Besides that, as traffic packets can be processed fast at LSR, label switching operation results in shorter jitter than with traditional IP routing.

When packets arrive at ingress LER, packets that has the same attributes (e.g. destination, quality of service and etc.) are classified into FECs. All packets that belong to a particular FEC will follow the same path. This allows packets of different classes be treated differently in MPLS networks. With constraint-based routing used in MPLS, packets in the same class can be routed with the required QoS fulfilled.

2.3.2 Traffic Engineering Support

Traffic engineering attempts to optimize users' QoS needs by making the best use of network resources to support those needs. Traffic should be routed through a given network in the most efficient and reliable manner.

Multiprotocol label switching with its efficient support of explicit routing provides basic mechanisms for facilitating traffic engineering. Constraint-based routing can be implemented easily in MPLS networks. Traffic engineering in MPLS controls traffic in a network using Constrained Shortest Path First (CSPF) instead of using the Shortest Path First (SPF). CSPF creates a path that takes restrictions into account. The chosen path will utilize links that are less congested. Therefore, MPLS traffic engineering allows traffic to be distributed across the entire network infrastructure. If the traffic load between a pair of ingress and egress routers exceeds the capacity of any single path, then the load can be split and multiple LSPs can be set up. Multipath traffic engineering can effectively control the network resource utilization.

By monitoring the traffic in a LSP tunnel, network operators can characterize end-to-end traffic flows within the MPLS domain. Traffic losses can be estimated by monitoring ingress LER and egress LER traffic statistics. Traffic delay can be estimated by sending probe packets and measuring the transit time. With these traffic data, service providers can make changes and improvements when necessary so that the Service Level Agreement (SLA) with users are met and resources are optimized.

At the same time, MPLS supports the concept of protection switching and backup paths to provide reliable data transmission. In case of a link or node failure in MPLS networks, affected traffic can be restored rapidly through backup resources. More issues on network survivability in MPLS networks are discussed in Chapter 3.

2.3.3 Multi-protocol Support

MPLS is applicable to any network layer protocol. Different protocols (e.g. IPv4, IPv6, IEEE 802.3 (Ethernet), VLAN, IEEE1394(DV)) are only distinguished at ingress and egress LERs. At the ingress LER packets are classified into FECs and label is pushed onto the label stack. LSRs along the LSP will forward the labeled packets based on the top label in label stack only. At the egress LER, label in the label stack will be removed. The packet will be forwarded to its next hop based on the packet's original contents, using the original protocol.

CHAPTER 3

Network Survivability Techniques in MPLS Networks

3.1 Introduction

As the telecommunication market is evolving towards services, network management becomes increasingly complex. Network survivability becomes one of the major issues in Service Level Agreements (SLA) as applications in the Next Generation Networks require high bandwidth connections that are reliable and at the same time optimize bandwidth resources utilization. Quality of service (such as bandwidth, delay and reliability) is required to provide the applications (e.g. multimedia, unified messaging and other e-commerce services) with better transmission performance than "best effort" service.

In case of providing reliability, the networks must be able to recover from link (or node) failure within the required period with bandwidth requirements fulfilled. To restore data transmission in case a link (or node) failure occurs, the affected data should be quickly re-routed. There are two main methods for network survivability in MPLS networks: protection and restoration. Details of these two methods are given in the following sections.

3.2 Protection

In protection scheme, backup resources (e.g. bandwidth) are reserved at the time of primary connection setup. That is, to route a label switched path (LSP) setup request successfully, both the primary LSP and backup LSP have to be set up at the same time. Traffic on primary LSP can be re-routed through the backup LSP without further delay once the failure on primary LSP is known by the ingress Label Edge Router (LER).

Based on the ways primary LSP is protected, there are generally three types of protection models: path protection, segment protection and link protection.

3.2.1 Path Protection Model

In path protection, a backup path starting from the ingress LER to the egress LER is used to protect the whole primary path (see Figure 3·1). Therefore, in case of a failure, the ingress LER is responsible for switching over the traffic to backup path. The drawback of this method is that the failure message will need to be propagated back to the ingress LER before the ingress LER knows the failure and restore transmission through backup path. Besides that, the backup path has to be link disjoint with the primary path when protecting against link failure.

Path protection can be further classified into two types: failure-independent and failure-dependent. In failure-independent path protection, the backup path has to be link (node) disjoint with the primary path. This is to ensure that when failures happen on the primary path, the backup path will not be affected and traffic can always be rerouted successfully. On the other hand, in failure-dependent path protection, depend on the primary path link that fails, different path (not necessarily link/node disjoint with

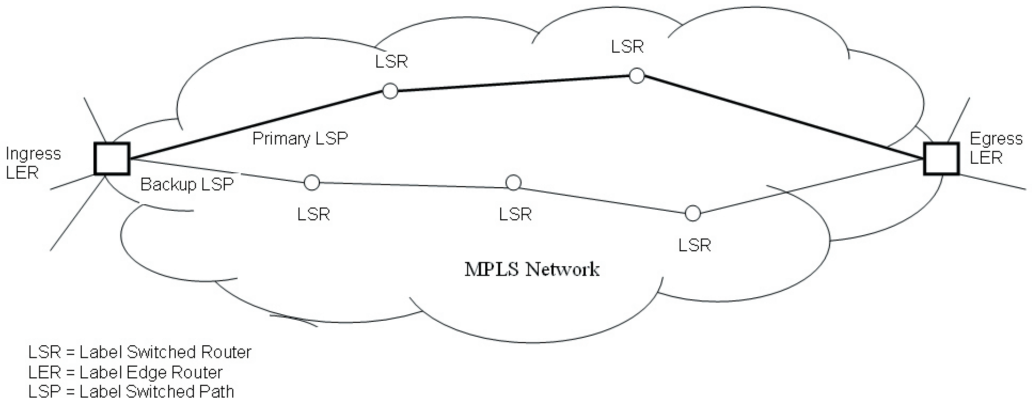

Figure 3·1: Path protection

the primary path) can be used. In both failure-independent path protection and failure-dependent path protection, the failure notification has to be sent to the source node as the source node of each LSP is responsible to reroute the affected traffic.

A lot of path protection methods for MPLS networks have been proposed in the literature, e.g. [4], [5], [6], [7], [23], [24]. In order to utilize the bandwidth resources efficiently, bandwidth sharing is adopted. Bandwidth on primary path cannot be share. However, it is possible that backup bandwidth reserved on a link be shared by multiple primary connections. For example, consider two LSPs between source LER, *s* and destination LER, *d*, each LSP requires *b* units of bandwidth. If the primary paths of these two LSPs are link disjoint and only a single link failure is considered, then it is impossible to have both the primary paths fail at the same time. Therefore, it is feasible for the two LSPs to share a backup path, and only *b* units of bandwidth are required on the backup path links. Sharing of backup bandwidth assumes that the primary LSPs that share their backup bandwidth will not fail at the same time.

It is found that the amount of sharing that can be achieved in the backup paths is a function of the information available to the routing algorithm [4]. When complete information is considered, it means that the routes for the primary and backup paths of all connections are known. This is only possible in centralized network control (see section 3.4.1) where a central network controller decides all the route setups. With complete information available, the number of backup paths bandwidth sharable among link-disjoint primary paths is optimized. When only the residual (available) bandwidth is known from each link, it is called the "no information" scenario in [4]. In "no information" scenario, no backup path bandwidth sharing can be done. Thus, bandwidth cannot be efficiently utilized.

In [4], a path protection method named Sharing with Partial Information (SPI) was proposed. By using the partial information (aggregated bandwidth used by primary path, aggregated bandwidth used by backup path and the link residual bandwidth) on a link, it is found that multiple link disjoint primary paths can share backup path with performance almost identical to that of the complete information model. Assume that the cost of link(*i,j*) on primary path is a_{ij}, the cost of link(*u,v*) on backup path is c_{uv} and the bandwidth required by the new connection is *b*. The cost of primary path is the sum of the a_{ij} for all links on the primary path and the cost of backup path is the sum of the c_{uv} for all links on the backup path. The objective of protection algorithm is to find the least cost primary and backup paths pair. To calculate the backup link cost c_{ij}, *M*, the largest value of aggregated primary bandwidth for some link(*i,j*) on the primary path have to be found.

$$c_{uv} = \begin{cases} 0 & \text{if } M + b \leq G_{uv} \\ M + b - G_{uv} & \text{if } M + b > G_{uv} \text{ and } R_{uv} \geq M + b - G_{uv} \\ \infty & \text{Otherwise} \end{cases} \tag{3.1}$$

where G_{uv} is the aggregated bandwidth used by backup paths on link(*i,j*) and R_{uv} is the link residual bandwidths on link(*i,j*). This problem is NP-hard. A linear programming model of the problem was presented in the paper [4].

A Distributed Partial Information Management (DPIM) model using only partial information is discussed in [5] and [6]. In DPIM model, each node *n* maintains link state information for each local link *e*. The link state information includes total primary bandwidth on link *e*, total backup bandwidth on link *e*, residue bandwidth on link *e*, and two vectors showing the profiles of primary bandwidth and backup bandwidth ($\mathcal{V}_{\mathcal{P}}(e)$ and $\mathcal{V}_{\mathcal{B}}(e)$). $\mathcal{V}_{\mathcal{P}}(e)$ and $\mathcal{V}_{\mathcal{B}}(e)$ contain additional information that is not utilized by SPI. $\mathcal{V}_{\mathcal{P}}(e)$ shows the amount of primary bandwidth on link *e* that is protected on other links in the network. $\mathcal{V}_{\mathcal{B}}(e)$ shows the amount of backup bandwidth reserved on link *e* that is used to protect primary connections on other links. The maximum bandwidth (V_{P_e}) obtained from the $\mathcal{V}_{\mathcal{P}}(e)$ entries gives the sufficient amount of bandwidth that needs to be reserved on any other links in the network in order to protect against the failure of link *e*. On the other hand, the maximum bandwidth (V_{B_e}) obtained from the $\mathcal{V}_{\mathcal{B}}(e)$ entries gives the minimum (or necessary) amount of backup bandwidth needed on link *e* to backup primary connections on other links. To find the link cost c_{uv} as in equation 3.1, the value of *M* need to be modified. In DPIM, the largest value of bandwidth (M') required to protect primary connections on primary path is equal to finding the largest value of V_{P_e} among all links *e* on the primary path.

$$M' = \max_{e} V_{P_e} \tag{3.2}$$

DPIM improves backup bandwidth sharing efficiency in a network. This is because M'

is smaller than M; as a result, better estimation of path costs can be obtained to help DPIM makes efficient routing decision in the network.

In [6], a heuristic algorithm was proposed to find a pair of link (or node) disjoint paths for each online request instead of using the time-consuming ILP model. The basic idea is to attach a potential backup cost that is derived mathematically to each link so that one may select a primary path when still taking into consideration the impact of bandwidth sharing along a yet-to-be-chosen backup path. Although the algorithm proposed can be used with only partial information under distributed control, it can also be applied under centralized control when complete information is available. The algorithm performance results show that it can have better overall performance than the time-consuming ILP in the online case.

Besides considering the total costs of both primary path and backup path, the backup path length (in term of hops) should be limited too. Long backup path will affect the restoration time and also the signal transmission quality. In [7], integer linear programming (ILP) models using two parameters (ϵ and μ) in Sharing with Complete Information (SCI) scheme and DPIM, were proposed to improve the network resource utilization and to keep the backup paths as short as possible.

When no sharing is possible (due to insufficient link state information), bandwidth efficiency has to be achieved through good path selection. The min-hop algorithm (MHA) [25] and the widest shortest path algorithm (WSP) [26] are two of the most used routing algorithms in the literature. The min-hop algorithm uses shortest path algorithm (e.g. Dijkstra's algorithm) to route connection along the path with minimum number of feasible links. Widest shortest path algorithm chooses a feasible minimum-hop path that has the largest free residual bandwidth at bottleneck link. Widest shortest path algorithm works better than min-hop algorithm as it tries to balance traffic load among feasible shortest paths.

In [27], two algorithms were presented for restorable routing without sharing and one of them using only shortest path computations. The objective is to improve performance by routing using the minimum interference criteria. The only link state information needed in [27] is the link residual bandwidth and knowledge of network ingress-egress pairs.

3.2.2 Segment Protection Model

Segment protection means that a primary LSP is divided into several segments and each segment is protected by a backup path (see Figure 3·2).

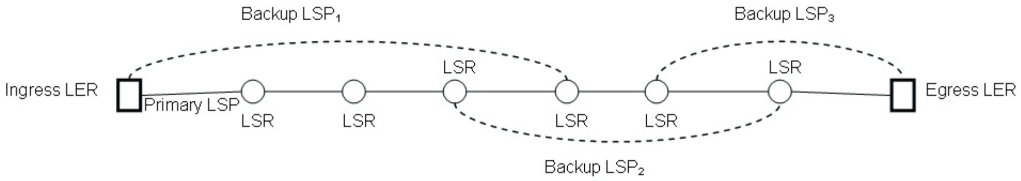

Figure 3·2: Segment protection

If link (or node) failure happens within a primary path segment, the failure message will be propagated back to the LSR at the beginning of that particular segment. The router LSR at the beginning of the segment is responsible to switch the traffic to backup path.

An innovative approach called PROtection using Multiple Segments (PROMISE) was proposed in [8]. In PROMISE, the backup path capacity can be shared in two levels: intra-demand sharing and inter-demand sharing. In intra-demand sharing [24], the backup capacity can be shared by backup LSPs belonging to the same primary LSP. On the other hand, inter-demand sharing refers to the sharing of backup capacity on a link-by-link disjoint primary LSPs. Therefore, PROMISE can provide bandwidth efficiency as good as the path protection, while at the same time recovering faster than path protection. An ILP model was developed in [8] to determine an optimal set of segments to protect a give primary path. The ILP approach is too time-consuming for large networks, therefore a fast heuristic algorithm based on dynamic programming was designed to obtain a near-optimal set of segments [8]. The heuristic algorithm proposed can achieve bandwidth efficiency as high as some shared path protection schemes and at the same time, much faster recovery than these shared path protection schemes.

In [11], a segment based algorithm, which provides efficient recovery from failure and guarantees QoS (e.g. end-to-end delay, jitter), was developed. QoS constraint e.g. bounded switch-over time (*Note: switch-over time is the time for which the packets will be dropped over the primary LSP after a failure*), is used when finding the backup LSP segments. The process of finding the backup path is combined with the process of segmenting the primary path. First, starting from the egress LER, the largest possible segment towards the ingress LER, which satisfies the bound on switch-over time, is found. Then, from the LSR at the beginning of the segment, a backup path needs to be set up to protect this segment. If no such backup path can be established, the segment size will be shortened by one link and a backup path corresponds to this new segment must be set up. This algorithm will produce segments that satisfy the switch-over time constraint, while at the same time minimize the number of segments required. It is said that multi QoS constraint satisfaction algorithms can also be developed using the algorithm developed in [11].

3.2.3 Link Protection Model

Link protection means that whenever a link (or node) failure happens, the first node upstream from the failure must be able to switch the affected traffic to the backup path (see Figure 3·3). For example, when a link(*i,j*) on primary path fails, the backup path for link(*i,j*) can be any path connecting nodes *i* and *j* that does not include link(*i,j*). Even though it is known that link protection is less bandwidth efficient than path protection, link protection provides faster failure resilience.

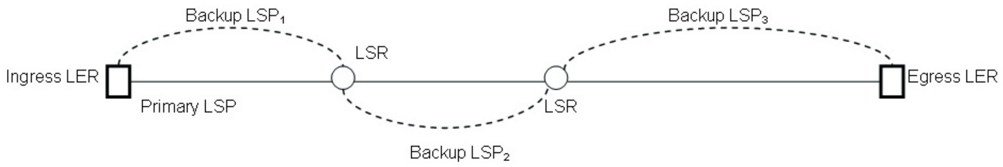

Figure 3·3: Link protection

A new QoS routing problem which requires the on-line routing of a bandwidth guaranteed path along with the setting up of backup paths for every link or node traversed by the primary path is considered in [9]. The paper shows that a partial information scenario, which uses only aggregated information, provides sufficient information for efficient dynamic routing of locally restorable bandwidth guaranteed paths. An efficient dynamic routing algorithm for bandwidth guaranteed paths that are locally restorable under single link or node failure was proposed in [9]. The routing is done using a sequence of shortest path computations. The routing objective is to minimize the amount of bandwidth used by primary and backup paths while protecting against single node or link failure. Intra-demand sharing together with inter-demand sharing of bandwidth is used to reduce the backup bandwidth reserved and thus improves the bandwidth efficiency.

3.3 Restoration

Restoration schemes search for unreserved network spare resources only after failure occurs. Therefore, protection methods recover faster from a failure than restoration schemes. However, as backup resources are reserved at the point of primary path setup, the protection schemes are less bandwidth efficient than restoration schemes. Also, due to restoration nature that looks for unreserved spare resources only after failure occurrence, restoration schemes cannot guarantee the recovery time, and the amount of information lost for real-time applications. This causes restoration schemes inappropriate for mission-critical applications. Restoration schemes are more applicable to highly dynamic networks with distributed control [10].

In [10], the concept of Differentiated Reliability (DiR) was extended to restoration schemes in which network resources for a disrupted connection are sought upon failure occurrence. The DiR concept is applied in two dimensions, restoration blocking probability and restoration time. Connections from high reliability classes are guaranteed lower restoration blocking probability and shorter restoration time than the connections from lower reliability classes. Differentiation in the two dimensions is accomplished by proposing three preemption policies that allow high priority connections to preempt resources allocated to low priority connections [10]. Results show that the proposed preemption policies can guarantee differentiated classes of reliability in terms of both restoration blocking probability and restoration time. It was found in [10] that by carefully choosing the preemption policy, the desired reliability degree can be obtained while minimizing the number of preempted connections.

3.4 Networks Control Models

As the amount of sharing that can be achieved in the backup paths is a function of the information available to the routing algorithm [4], in order to get good performance from the protection schemes, it is important to know the amount of network traffic information that can be obtained. Based on the way network resources are controlled, network control models are classified into two types: centralized and distributed. When the network resources are controlled by a single central controller, it is called centralized control networks. On the other hand, in the case where each router performs routing locally, the networks are called distributed control networks.

3.4.1 Centralized Control Networks

In centralized control network, all connection setup requests are forwarded to the central controller. All routing decisions are made by one central computer or router. Therefore, the central controller knows all the routes for the primary and backup paths of all connections currently in progress. Complete routing information can be obtained easily in centralized control networks to get optimal bandwidth usage in backup bandwidth sharing. However, centralized control networks are bad in scalability as route computations needed for large networks are enormous.

3.4.2 Distributed Control Network

In distributed control network, all routers in the network make their own routing decisions. Each node maintains only routing information of connections that traverse through it. Distributed control networks can provide partial information that may include aggregated bandwidth used by primary path, aggregated bandwidth used by backup path and the link residual bandwidth on a link. These information can be disseminated to all the nodes in distributed control networks using routing protocols and traffic engineering extensions.

CHAPTER 4

Multipath Protection in MPLS Networks

In this chapter, we develop a multipath protection scheme for MPLS networks. Here, a single path is used as primary path and multiple paths are used as backup paths. All the backup paths are link disjoint with the primary path. Multipath routing is used to reduce the load on the congested links in a network and to improve backup bandwidth sharing. Efficient multipath protection algorithms are proposed. Simulations performed show that the proposed algorithms achieve better network efficiency than single-path protection scheme.

4.1 Introduction

In order to send data in MPLS networks, an explicit path needs to be established between the source and destination nodes. Bandwidth required on the path is reserved for the connection until it is terminated. However, by using only one path when transmitting data between source and destination routers, network resources cannot be utilized efficiently. Congestion in networks can be reduced by using optimal routing. In optimal routing, traffic load between a source and destination pair can be sent through more than one path. In fact, optimal routing is based on the sophisticated mathematical theory of optimal multi-commodity flows [28].

In multipath routing, several paths can be used to send data from a source router (ingress LER) to a destination router (egress LER). As a result, multipath routing allows the traffic loads in a network to be spatially distributed over all links. This load-balancing technique reduces load on bottleneck links leading to improved resource utilization.

The problem of finding a primary path and multiple backup paths that minimize the bandwidth consumed is NP-complete [29], [30]. Further, it is not desirable to reroute the existing demands in a network whenever a new demand arrives, in order to optimize the total bandwidth used in the network. Therefore, efficient heuristic algorithms that do not require to change the existing demands to honor a new demand, are needed for large networks.

Two multipath protection heuristic algorithms in dynamic MPLS networks are proposed in this chapter. Instead of knowing all the demands in advance, demand requests in dynamic MPLS networks arrive and depart in a random manner. Routing decision is done online when a request arrives; therefore, the routing algorithms must be fast and efficient.

In the multipath protection algorithms proposed, a single path is used for primary path, and multiple paths are used as backup paths. The reason of doing so is that, primary path transmission is more significant. Therefore, the best performance is achieved by using single path (so no delay incurred by multipath routing). When multiple paths are used, packets travel along different paths may arrive out of order and have to be resequenced. To avoid this problem, ingress routers should provide a flow-level forwarding mechanism. The partitioning of a traffic demand can be done by adjusting output range of the hashing function of dynamically changing traffic flows [31], [32].

4.2 Motivation

Even though the topic of multipath routing has been discussed in previous works [14], [33], [20], [34], [15], [17], [18], it is found that, those works focus on solving the multipath routing problem for primary connections in offline scenario where all the traffic demands are known. Integer linear programming (ILP) is usually used to do offline routing. However, due to the computational complexity in obtaining an optimal solution using integer linear programming, efficient heuristic algorithm is more desirable for practical use. Moreover, when multiple paths are used for the routing of a primary connection, it may require more total bandwidth than the single shortest path. If multiple paths are used as backup paths instead of primary paths, and if backup bandwidth sharing is possible, the amount of bandwidth wasted can be reduced so that more future connections can be accommodated.

To accept a connection setup demand with path protection, primary and backup paths must be set up at the same time. When single backup path scheme is used, if a single backup path cannot be found due to bottleneck links, the demand will be rejected. When multiple backup paths are used, the amount of backup bandwidth required on each path is reduced; therefore, it is more likely that multiple backup paths with enough bandwidth can be found. For example, in Figure 4·1, there are 8 undirected links. Each undirected link in Figure 4·1 represents two directed links. If a new setup request that requires 9 units of bandwidth arrives at node 1 and its destination is node 3, a primary path ($1 \rightarrow 5 \rightarrow 3$) can be set up to route the traffic. However, using the single backup path algorithm, the request has to be dropped because no link-disjoint backup path that satisfies the 9 units backup bandwidth requirement can be found. If multiple backup paths are allowed, backup bandwidth required can be split into two parts (e.g. 5 and 4) and routed through path $1 \rightarrow 2 \rightarrow 3$ and path $1 \rightarrow 4 \rightarrow 3$.

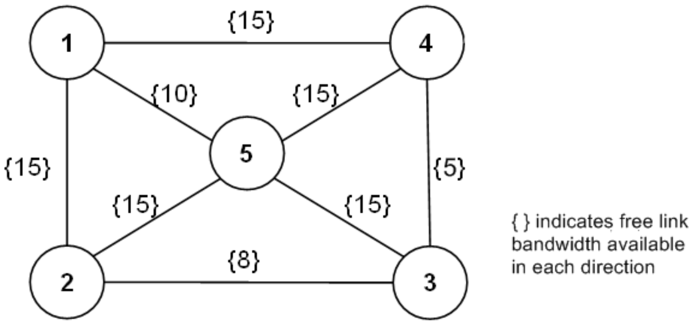
{15}
{10}
{15}
{15}
{5}
{15}
{15}
{8}
1
4
5
2
3
{ } indicates free link bandwidth available in each direction

Figure 4·1: Example network

In addition to that, multiple backup path routing can further improve the performance of backup bandwidth sharing. By allocating the backup bandwidth needed on separate link-disjoint paths, backup bandwidth reserved on each backup path is smaller. As a result, better bandwidth sharing can be achieved. For example, using the same network in Figure 4·1, two requests (r_1 and r_2) as given in Table 4.1 are to be routed.

Table 4.1: Example of two setup requests

Demand ID	Source	Destination	Bandwidth
r_1	1	3	6
r_2	2	4	10

If single backup path algorithm is used (see Figure 4·2), no backup bandwidth sharing is possible. r_1 is routed through primary path $1 \rightarrow 5 \rightarrow 3$ and backup path $1 \rightarrow 2 \rightarrow 3$. Primary path of r_2 is $2 \rightarrow 5 \rightarrow 4$ and backup path of r_2 is $2 \rightarrow 1 \rightarrow 4$. r_2 cannot be routed on path $2 \rightarrow 3 \rightarrow 4$, as there are not enough bandwidth on the links. 64 units of bandwidth are used in the network.

On the other hand, if multiple backup paths are used, the total bandwidth used is reduced to 58 units. Total amount of bandwidth used is reduced by 6 units because backup bandwidth sharing is achieved with the application of multipath protection. One primary path and two backup paths are used by each demand. Primary path of r_1 is $1 \rightarrow 5 \rightarrow 3$. Path $1 \rightarrow 4 \rightarrow 3$ and path $1 \rightarrow 2 \rightarrow 3$ are the backup paths of r_1, 3 units of bandwidths are reserved on each of the backup path. Primary path of r_2 is $2 \rightarrow 5 \rightarrow 4$. Path $2 \rightarrow 1 \rightarrow 4$ and path $2 \rightarrow 3 \rightarrow 4$ are the backup paths of r_2, 5 units of bandwidths are reserved on each of the backup path. As a result, r_1 and r_2 can share backup bandwidth on link(1,4) and link(2,3).

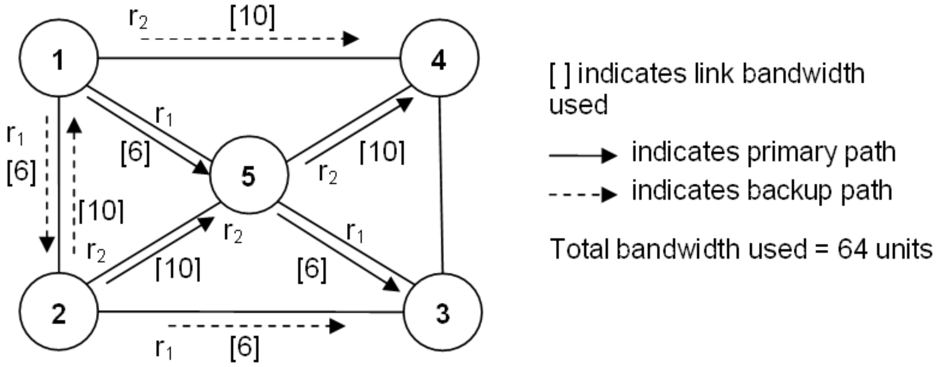

Figure 4·2: Routing of demands with single backup path

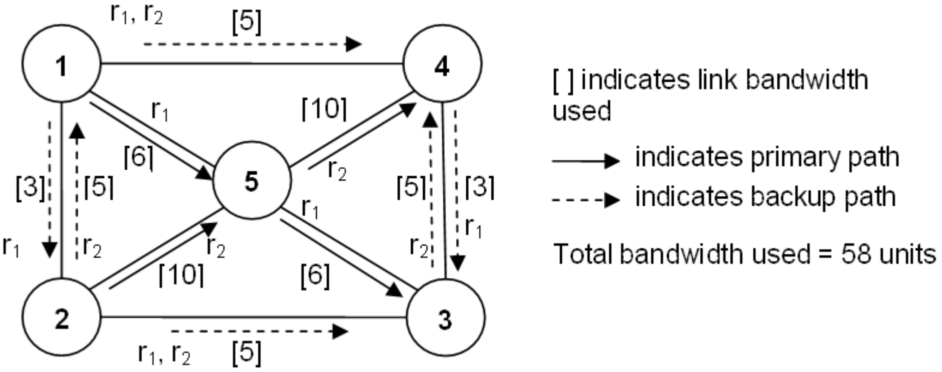

Figure 4·3: Routing of demands with multiple backup paths

Besides that, by using traffic engineering techniques (e.g. load balancing) in multipath protection, the loads on different links of the network can become as balanced as possible; thus, prevent critical network resources from being exhausted early and becoming a bottleneck.

4.3 Related Work

Considerable work has been done to analyze the performance of multipath routing in networks. It has been shown in [13] that the connection establishment time for reservation is significantly lowered when using multipath routing. In [33], the paper aims to determine the traffic split ratio to minimize the end-to-end delay and loss rate for differentiated services. The use of a dynamic traffic partitioning and assignment methodology to adaptively map ingress traffic into several parallel LSPs, were investigated in [33]. A stochastic framework for the traffic partitioning problem were presented. Within the framework, a set of parallel link disjoint LSPs is modeled by parallel queues and a partitioning algorithm is devised for different service classes that is adaptive to the prevailing state of the network.

Two traffic congestion control techniques: flow assignment and packet scheduling, were investigated in [35] to route packets efficiently in multipath networks. The flow assignment mechanism is used to find optimal splitting of traffic on multiple paths. The packet scheduling mechanism is utilized to reduce the packet re-sequencing delay and the consumption of re-sequencing buffer in networks. A multiple node *M/M/1* tandem network with a delay line is considered as the path model in this paper. The analytical results show that when end-to-end path delays are Gaussian distributed, the techniques proposed are very effective in reducing the average end-to-end path delay, the average packet re-sequencing delay and the average re-quencing buffer occupancy for various path configurations.

A constrained multipath traffic engineering scheme for MPLS networks was proposed in [15]. In the paper, constraints such as the maximum hop count constraint and preferred or not-preferred node/link list are considered in ILP algorithms that calculate explicit paths and split ratio offline are presented. The paths and split ratio that are obtained after solving the ILP proposed are passed to MPLS edge routers for explicit LSP setup. For easy implementation at the routers, only discrete values are used for the split ratio.The experiment results show that the proposed algorithms are fast and superior to the conventional shortest path algorithm in terms of maximum link utilization, total traffic volume and number of required LSPs.

In [16], a fault-tolerant multipath traffic engineering scheme was proposed. This scheme includes the maximally disjoint multipath configuration and the traffic rerouting mechanism for fault recovery. The hop-count and path-count constrained maximally disjoint multipath configuration problem is defined by ILP formulation. When the statistical traffic demand is known, the maximally disjoint multipath that minimizes the maximum link utilization is found using the ILP solver such as CPLEX, to satisfy every given demands. When link failed, the traffic on the failed LSPs is required to be rerouted into other available LSPs. The traffic rerouting problem is formulated as LP problem; therefore, it need to be solved by LP solver when link failure occurs at real-time.

A state-dependent traffic engineering mechanism called Multipath Adaptive Traffic Engineering (MATE) was proposed in [14]. MATE is targeted for switched networks such as MPLS networks. The main goal of MATE is to avoid network congestion by adaptively balancing the load among multiple paths based on measurement and analysis of path congestion. MATE is intended for traffic that does not require bandwidth reservation. Optimization decision in MATE is done based on path congestion measure obtained using probe packets. The traffic engineering function comprises two phases: a monitoring phase and a load-balancing phase. If appreciable and persistent change in the network state is detected in monitoring phase, transition is made to the load-balancing phase. In the load-balancing phase, MATE algorithm will try to equalize the congestion measures among the LSPs. Analytical models were provided to prove the stability and optimality of MATE. It is found that, in many cases, high packet loss rates can be significantly reduced by properly shifting some traffic to less loaded LSPs.

A traffic engineering scheme using multiple multipoint-to-point (m-t-p) label switched paths (LSPs) which can reduce the number of LSPs and required labels in links was proposed in [20]. The scheme includes the m-t-p LSP creation and flow assignment. After routes are selected, m-t-p LSPs are designed to include them. M-t-p LSPs are created simply based on the network topology alone. The m-t-p LSP design problem is formulated as an integer programming problem. The m-t-p LSPs created should satisfy the requirement that each ingress/egress node pair have a diversity of routes including at least one route that is not affected in each individual failure case. The flow assignment problem is formulated as a mixed integer programming problem in which maximum link load, i.e., maximum congestion, is minimized. A backup route is prepared for each working route. The link capacities along backup routes are required only when their working routes are damaged. Numerical examples show that the proposed flow assignment scheme can reduce maximum link load in comparison with the shortest path fast based flow assignment and achieved effective load balance across each example network.

Fast heuristic algorithms are used in [17], [18] and [34]. In [17], the traffic engineering mechanism proposed is called Load Distribution over Multipath (LDM). LDM aims to enhance the network utilization as well as the network performance by adaptively splitting traffic load among multiple paths. First, a set of paths is pre-computed between every source and destination pair. After that, a set of candidate LSPs to be used in the traffic distribution are selected based on the current network state information. Routing decisions are made at the low level and the traffic divided into each path is proportional to the length and the load of a path. LDM tries to expand the candidate path set if the congestion level of the candidate LSP set increases. LDM is intended for the best-effort type traffic that does not impose any specific service requirement to the network.

Two constraint-based routing algorithms were proposed in [18]. If no single path satisfying the constraint can be found, the algorithms proposed in [18] can divide the bandwidth constraint into two or more sub-constraints and find a constrained path for each sub-constraint. The first algorithm used is called equal bandwidth multi-path constraint-based routing algorithm. This algorithm divides the bandwidth requirement into multiple sub-constraint with equal bandwidth and the process continues until multiple paths whose total path bandwidth is equal to or greater than the bandwidth requirement are found. The second algorithm, which is called maximum path bandwidth first multi-path routing algorithm, tries to find minimum number of multiple paths. The algorithm first tries to compute a single path that can satisfy the constraint. If no single path can be found, it computes the path with maximum bandwidth available and allocates bandwidth to the path. The algorithm then computes another constrained path using the remaining bandwidth constraint. The algorithm continues until it allocates all the bandwidth required. The simulation results show that the second algorithm needs less number of paths; however, it is also found that the second algorithm utilizes more bandwidth resource than the first algorithm does for the same condition.

In [34], a heuristic algorithm for hop-count and path-count constrained dynamic multipath routing was proposed. This heuristic aims to minimize the maximum of link utilization. By computing the constrained multiple paths and their split ratios in polynomial time, the proposed heuristic approximates the traffic bifurcation mixed integer programming (MIP) problem that is NP-hard. M multiple paths are selected using the M shortest path algorithm [36]. After finding the M multiple paths, the amount of a traffic demand is divided to M paths using the load split algorithm defined. The traffic split ratio obtained is based on traffic partitioning by hashing at flow level.

4.4 Problem Definition

Multipath algorithms presented in Section 4.3 aim to solve the multipath routing problem of primary connections. Here, the proposed algorithms use a primary path and multiple backup paths for every LSP setup request accepted in order to provide failure-independent path protection.

Consider a MPLS network $\mathcal{G}$ with $\mathcal{N}$ nodes and $\mathcal{E}$ undirected links. The $\mathcal{N}$ nodes include ingress routers, egress routers and core (transit) routers. LSP setup requests arrive one by one and every LSP terminates after certain duration. A LSP setup request (s, d, b) is defined by its source (s), destination (d), and the requested bandwidth (b). For each LSP setup request to be accepted, a primary path and one (or more) backup path have to be set up. Backup bandwidth sharing is allowed to minimize the redundancy caused by backup protection. The primary path should be link-disjoint with the backup path to protect the LSP from single link failure. Only single link failure is considered here.

When a connection request arrives at an edge node, the routing decision can be made locally at the edge node (as in a distributed network) or the request can be forwarded to a central controller (as in a centralized network). In the case of a distributed network, each edge node is assumed to have limited knowledge (partial information) on the traffic situation of every links, only the amount of free bandwidth, the amount of bandwidth used by primary connections and the amount of bandwidth reserved for backup paths are known. Link state information, which includes partial information, is disseminated using traffic engineering extensions. While in centralized network, complete routing information of primary path and backup paths of every connection in the networks are known. The amount of traffic information available when making routing decision for connection requests will affect the amount of backup sharing on each link [4].

When partial information is available in network, the amount of free residual bandwidth on link(*i,j*) denoted by F_{ij}, the amount of bandwidth used by primary connections on link(*i,j*) denoted by P_{ij}, and the amount of bandwidth reserved for backup paths on a link(*i,j*) denoted by B_{ij}, are used to calculate the link cost of backup path. Assumed that the primary path is determined before searching for backup path, the cost of backup path link(*u,v*) is

$$c_{uv} = \begin{cases} 0 & \text{if } M + b \leq B_{uv} \\ min(M + b - B_{uv}, b) & \text{if } M + b > B_{uv} \text{ and } F_{uv} \geq M + b - B_{uv} \\ \infty & \text{Otherwise} \end{cases} \tag{4.1}$$

where

$$M = \max_{(i,j)} P_{ij} \tag{4.2}$$

for every link(*i,j*) selected on primary path. M represents the largest value of P_{ij} for some link(*i,j*) on the primary path. For a potential backup path link(*u,v*), if $M + b \leq B_{uv}$, it means any link failing on the primary path require at most $M + b$ unit of bandwidth on the links of the backup path. Therefore, no additional bandwidth needs to be reserved on link(*u,v*) as the backup bandwidth B_{uv} on link(*u,v*) is enough to protect the primary path. If $M + b > B_{uv}$, then only B_{uv} unit of bandwidth are shareable, additional reservation of bandwidth ($min(M + b - B_{uv}, b)$) need to be made. $min(M + b - B_{uv}, b)$ tells that the maximum amount of additional bandwidth needed is b. If sufficient free bandwidth cannot be found, then link(*u,v*) is not feasible.

In centralized network with complete information, σ_{ij}^{uv} the amount of primary bandwidth on link(*i,j*) that is protected on link(*u,v*) is known. The cost τ_{ij}^{uv} of using link(*u,v*) on the backup path if link(*i,j*) is used on the primary path must be found.

$$\tau_{ij}^{uv} = \begin{cases} 0 & \text{if } \sigma_{ij}^{uv} + b \leq B_{uv} \text{ and } \textit{(i,j)} \neq \textit{(u,v)} \\ min(\sigma_{ij}^{uv} + b - B_{uv}, b) & \text{if } \sigma_{ij}^{uv} + b > B_{uv} \text{ and } F_{uv} \geq \sigma_{ij}^{uv} + b - B_{uv} \\ & \text{and } \textit{(i,j)} \neq \textit{(u,v)} \\ \infty & \text{Otherwise} \end{cases} \tag{4.3}$$

τ_{ij}^{uv} tells the amount of additional bandwidth needed on potential backup path link(*u,v*) to protect connections on primary path link(*i,j*). The value of τ_{ij}^{uv} is bounded by b as b unit of bandwidth is sufficient to protect the connection of LSP setup request (*s*, *d*, *b*). The actual link cost c'_{uv} used to find backup path in the network is

$$c'_{uv} = \max_{(i,j)} \tau_{ij}^{uv} \tag{4.4}$$

for every link(*i,j*) selected on primary path. c'_{uv} shows the amount of additional bandwidth required on link(*u,v*) in order to protect connections on every link of the primary path.

4.5 Algorithms Description

To find primary path and multiple backup paths that minimize the amount of bandwidth consumed is NP-Complete [29], [30]. Using integer linear programming (ILP) to find the routes in a large network is undesirable; therefore, heuristics to find a primary and its backup paths are developed.

In the multipath protection proposed, the active path first heuristic [37] is used. In the active path first heuristic, for every connection request, a shortest path with minimal number of links is found to be the primary path. Only after a primary path is found, backup path is selected from the network.

A primary path can be found using any shortest path algorithm (e.g. Dijkstra's algorithm, Bellman-Ford algorithm). Then, a new graph without the links of primary path can be formed. Every link in the new graph is assigned a link cost *c*. *c* is the amount of bandwidth that needs to be reserved on the link to protect the primary path. To improve bandwidth utilization efficiency, backup path found using link cost *c* is the path that can share most backup bandwidth.

Multiple link-disjoint backup paths are suggested to protect a connection. For example, if maximum *m* backup paths are allowed to be used to protect a primary path, these *m* link-disjoint backup paths can be found one by one using shortest path algorithm. By using only shortest path algorithm, primary and backup paths can be found in very short duration.

Since a connection is protected using m backup paths, the amount of bandwidth allocated on each backup path is less than the bandwidth required on primary path. Based on the ways of splitting traffic load on backup paths, two multipath protection heuristics are proposed in the following sections. For both the proposed heuristic algorithms, traffic partitioning can be done on a per flow basis by adjusting the output range of hashing function [31], [32].

4.5.1 Equal Splitting Multipath Protection

The easiest way of distributing backup bandwidth is to split the requested bandwidth equally to m backup paths. After a primary path is found, m link-disjoint backup paths are to be searched one by one using Dijkstra's shortest path algorithm. For example if $m = 2$, then the requested bandwidth b is divided by two. Each of the backup paths searched must fulfill the bandwidth requirement $b/2$.

Using the network describe in Section 4.4, for a setup request r with its source s, destination d, and requested bandwidth b, the equal splitting multipath protection algorithm works as follows:

1. Remove all the links e in $\mathcal{G}$ whose free link bandwidth is smaller than b to form $\mathcal{G}'$.

2. Use Dijkstra's algorithm to find a shortest path p_0 in $\mathcal{G}'$. p_0 is the primary path for request r.

3. Divide the bandwidth b into m smaller bandwidth b_k, $b_k = b/m$, $1 \leq k \leq m$.

4. For every b_k, $1 \leq k \leq m$, do the following steps to find path p_k.

a. For bandwidth b_k, remove all primary path p_0's links and all backup path p_j's links to form $\mathcal{G}''$, where $1 \leq j < k$.

b. Find a backup path p_k for bandwidth b_k using suitable link cost. The link cost in $\mathcal{G}''$ depends on the amount of traffic information available in the network. If complete information is used, then c'_{uv} at Section 4.4 on page 43 is the link cost. Else, with partial information, c_{uv} (at Section 4.4) is the link cost used. Bandwidth b_k is used to calculate backup link cost instead of b. Therefore, the value of c'_{uv} or c_{uv} obtained should be smaller than the value when b is used.

c. If no path can be found, reduce m by one and go to step 3, m must be larger than zero. If $m = 0$, it means no backup path can be found and request r must be dropped. Else if a backup path p_k is found, go to step 4d.

d. Record the backup path p_k. If $k = m$, go to step 5, else let $k = k + 1$ and go to step 4b.

5. Record the primary and backup paths found and update the bandwidth used on primary and backup paths links.

Equal splitting multipath protection is very time efficient because bandwidth is divided equally to every backup path. No sophisticated load balancing algorithm is required.

4.5.2 Failure Dependent Multipath Protection

To improve the network resource utilization, load balancing should be used to split the bandwidth for multiple backup paths. Instead of equally splitting the backup bandwidth and then find paths with sufficient bandwidth as backup paths, another method is proposed to improve the network performance.

First, *m* least cost paths which are link-disjoint with the primary path are selected. Then, based on the load on each backup path, the requested bandwidth is split so that the average amount of bandwidth that is reserved on backup path links to protect the primary path links is (approximately) identical. To further improve the load balancing performance, failure dependent scenario can be taken into consideration. By considering the failure dependent scenario, it means that for different primary path link failures, different amount of bandwidth is allocated on each backup path.

For example, when two backup paths are used ($m = 2$) for a demand with *b* unit of bandwidth, assume that g_i is the average amount of bandwidth that was previously reserved on backup path 1 to protect link *i* failure on primary path, h_i is the average amount of bandwidth that was previously reserved on backup path 2 to protect link *i* failure on primary path. If x_i is the fraction of *b* bandwidth split to backup path 1 and y_i is the fraction of *b* bandwidth split to backup path 2 when link *i* fails, then after using the load balancing method proposed, $x_i \times b + g_i \approx y_i \times b + h_i$. If the primary path found consists of *n* links, then there are *n* different backup bandwidth allocations on the backup paths.

After the load balancing step, the primary and backup paths can be set up. The backup paths' information will be sent to every node on the primary path and primary path's information will be sent to nodes on backup paths.

Using the network describe in Section 4.4, for a LSP setup request *r* with its source *s*, destination *d*, and requested bandwidth *b*, the failure dependent multipath protection algorithm works as follow:

1. Remove all the links *e* in $\mathcal{G}$ whose free link bandwidth is smaller than *b* to form $\mathcal{G}'$.

2. Use Dijkstra's algorithm to find a shortest path p_0 in $\mathcal{G}'$. p_0 is the primary path for request *r*.

3. Remove all primary path links from $\mathcal{G}$ to form $\mathcal{G}''$.

4. In $\mathcal{G}''$, find *m* link-disjoint shortest path p_k, one by one using Dijkstra's algorithm, $1 \leq k \leq m$.

5. Go to step 6 if all *m* backup paths are found. If *m* link-disjoint backup paths cannot be found, reduce *m* by one. If *m* = 0, halt and drop request *r*. Else go to step 4.

6. For every link *i* on p_0, do the following:

   i. Find the average backup bandwidth $bw_k(i)$ on p_k to protect primary connections on link *i*, $1 \leq k \leq m$.

   ii. Bandwidth *b* is split into $b_1(i)$, $b_2(i)$, ..., $b_m(i)$, so that

$$bw_1(i) + b_1(i) \approx bw_2(i) + b_2(i) \approx \ldots \approx bw_m(i) + b_m(i) \qquad (4.5)$$

7. Record the primary and backup paths found and update the bandwidth used on primary and backup paths links.

4.6 Performance Study

Extensive simulations have been generated for the two heuristic algorithms proposed: failure dependent multipath protection and equal splitting multipath protection. One network with 15 nodes and 56 links (Network 1), and another network with 70 nodes and 206 links (Network 2), are used in the simulations. The 15 node network is shown in Figure 4·4 below, each link represents two unidirectional links connecting the two end nodes.

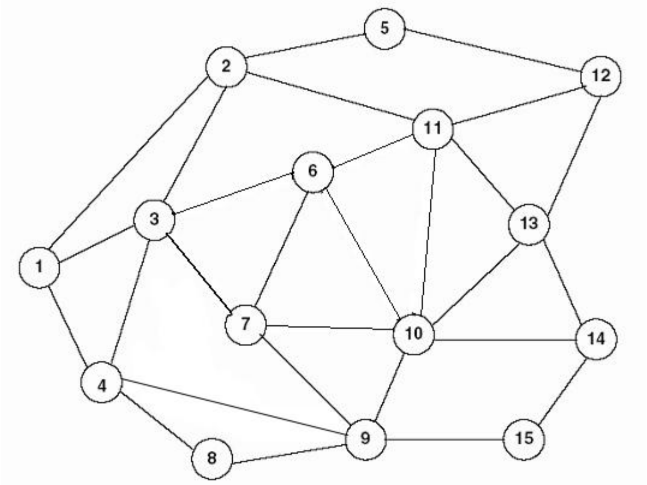

Figure 4·4: Network 1

Simulations were performed to study the behaviour of the algorithms with respect to the number of demands dropped when there is an overloading of the networks. An event-driven simulator written in C was used to generate the results.

Maximum two paths are selected as backup paths (m = 2) because too many backup paths may cause large delay when splitting the traffic. In Network 1, every link has link capacity equal to 40 units. The LSP setup requests arrive one at a time to the network. The source and destination for the LSP setup requests are selected randomly. Bandwidth required by LSP setup requests are uniformly distributed between 1 and 10 units. A setup request is dropped when there is not enough bandwidth capacity for either the primary path or the backup paths. 100 LSP setup requests are loaded to the network. The complete information and partial information scenarios are considered to study the performance of the proposed algorithms with different network information. The simulation is run for 10 different traffic patterns (10 different random seeds). The results showing the number of dropped requests for each of the 10 seeds is presented in Table 4.2 and Table 4.3.

Table 4.2: Number of dropped requests for 10 random seeds in Network 1 with partial information

Seed	Single Path Protection	Failure Dependent Multipath Protection $m = 2$	Equal Splitting Multipath Protection $m = 2$
1	19	16	17
2	11	10	13
3	18	15	17
4	14	12	13
5	22	18	19
6	10	10	9
7	6	5	4
8	21	16	17
9	9	8	8
10	15	12	14
Average	14.5	12.2	13.1

Table 4.3: Number of dropped requests for 10 random seeds in Network 1 with complete information

Seed	Single Path Protection	Failure Dependent Multipath Protection $m = 2$	Equal Splitting Multipath Protection $m = 2$
1	14	10	10
2	11	9	10
3	12	10	11
4	8	5	4
5	13	10	10
6	4	4	3
7	3	2	3
8	16	15	16
9	5	3	5
10	10	10	12
Average	9.6	7.8	8.4

In Table 4.2 and Table 4.3, the performance of the algorithms proposed is better than the single path protection. It is also found that failure dependent multipath protection performs slightly better than equal splitting multipath protection in both complete information and partial information scenarios.

Next, similar simulations on a larger network with 70 nodes and 206 links (Network 2) are performed. All links capacities are 40 units. A total of 1000 LSP setup requests are made for each simulation. Table 4.4 and Table 4.5 show the number of requests dropped for each of the 10 seeds.

Table 4.4: Number of dropped requests for 10 random seeds in Network 2 with partial information

Seed	Single Path Protection	Failure Dependent Multipath Protection $m = 2$	Equal Splitting Multipath Protection $m = 2$
1	291	254	257
2	267	240	238
3	283	240	249
4	305	260	259
5	274	254	253
6	298	268	264
7	298	261	270
8	291	257	264
9	246	219	216
10	301	258	277
Average	285.4	251.1	254.7

In Tables 4.4 and 4.5, the performance of equal splitting multipath protection and failure dependent multipath protection are better than the performance of single path protection.

Table 4.4 shows that failure dependent multipath protection performs slightly better than equal splitting multipath protection. In spite of that, the performance of failure dependent multipath protection is not better than equal splitting multipath protection in Table 4.5. The performance of failure dependent multipath protection is not always better than the performance of equal splitting multipath protection. This may be because in equal splitting it will always reduces congestion on a backup path link by reserving only

Table 4.5: Number of dropped requests for 10 random seeds in Network 2 with complete information

Seed	Single Path Protection	Failure Dependent Multipath Protection $m = 2$	Equal Splitting Multipath Protection $m = 2$
1	217	212	195
2	209	200	186
3	204	203	203
4	223	216	217
5	214	199	202
6	225	220	217
7	225	221	214
8	212	206	202
9	179	174	163
10	218	211	203
Average	218	206.2	200.2

half the bandwidth needed in primary path. However, in failure dependent multipath protection, the splitting of backup bandwidth may not be even on both paths, causing some links to be allocated more backup bandwidth. In addition to that, as online traffic is considered, routing decision that is made when a request arrives may cause some links to have congestion in the future since future demands are unknown. As a result, even though backup paths with least cost are chosen and load balancing are used to balance the bandwidth needed on both backup paths, failure dependent multipath protection does not outperform equal splitting multipath protection.

From simulation results we observe that the improvement of multiple backup path algorithms over single path protection is larger in partial information scenario than that in complete information scenario. This is due to the fact that, the routing decisions made with partial information are not as bandwidth efficient as the routing decisions made when all the connections in a network are known with complete information. With partial information, the amount of bandwidth required to protect primary connections is estimated at the routing decision making stage, causing inaccurate backup path cost to be used and more backup bandwidth to be consumed. When multipath protection algorithms are used, backup bandwidth sharing is improved; thus, more demands can be accepted. On the other hand, with complete information given, a routing decision is made to optimize network resource utilization based on the actual traffic distribution in a network. Since network resources are already utilized efficiently, this causes the improvement achieved by multipath protection algorithms to be less significant in complete information scenario.

4.7 Summary

The problem of online routing of LSPs with joint setting up of multiple backup paths for protection in MPLS networks has been studied in this chapter. The multipath protection algorithms proposed aim to utilize network bandwidth efficiently and at the same time reduce the number of LSP setup requests dropped because single backup path cannot be found. By using multiple backup paths, backup bandwidth sharing is also improved. This is because requested bandwidth is distributed into several paths, thus reduces the amount of bandwidth that needs to be reserved on every backup path link.

Two multipath backup protection schemes are proposed in this chapter. The equal splitting multipath protection scheme divides the required backup bandwidth of a request equally to multiple backup paths. It helps distributing traffic loads over the network and reduces congestion on bottleneck links. In failure dependent multipath protection, for every primary path link, allocations of bandwidth on backup paths are different.

Both algorithms proposed are fast and suitable for dynamic routing MPLS networks as they use shortest path algorithm to find the primary and backup paths required for a

LSP setup request. Through simulations performed, it can be found that the multipath protection algorithms utilize network resources more efficiently and thus allow more LSP setup requests to be accepted even in heavily-loaded network.

CHAPTER 5

Protection with Supplementary Information for MPLS Networks

In this chapter, several new supplementary information are proposed to further improve backup bandwidth sharing and network utilization in dynamic distributed MPLS networks. These supplementary information include primary bandwidth change, backup bandwidth change and source-destination pair traffic distribution. The supplementary information proposed can be disseminated in distributed networks using traffic engineering extensions protocols. Through the simulations performed, it is found that the supplementary information helps to reduce congestion and the blocking probability of requests in networks.

5.1 Introduction

Failure independent path protection approach is commonly used to protect time critical applications. Assume that only single link failure can happen at any given time, primary path should be link disjoint with the backup path so that when failure happens, only one path will be affected. As two explicit paths are set up for one connection, network resources are not efficiently used in normal situation. Backup bandwidth sharing approach is introduced in [4], [5], [6] to increase bandwidth utilization.

In fault-tolerant distributed MPLS networks, in order to provide backup bandwidth sharing, some aggregate link state information has to be disseminated. The information includes total bandwidth used by primary LSPs, total bandwidth used by backup LSPs and total free residual bandwidth on a link. Such information is called the partial information [4] as it does not have information about individual connection routes and bandwidths. As partial information can be easily maintained and distributed to every ingress router using traffic-engineering extensions, it is used as basic link state information in the path protection heuristic proposed.

Though several path protection schemes have been previously proposed in [4], [5], [6], it is found that these protection schemes do not take into consideration the traffic situation on the links in MPLS networks. In this chapter, additional network information to further improve backup bandwidth sharing and network utilization in dynamic distributed MPLS networks are introduced. The effect of considering primary bandwidth change, backup bandwidth change and source-destination pair traffic distribution in fault-tolerant MPLS networks is studied.

As the problem of finding minimum cost link-disjoint primary and backup paths in backup bandwidth sharing networks is NP-hard [38] (because link costs are different for primary path and backup path), efficient heuristic algorithm is needed. In addition to that, dynamic traffic that arrives one by one without priori knowledge is considered here; therefore, the algorithm presented must be an online algorithm. To fulfill all these requirements, a sensible heuristic algorithm using only shortest path computations is presented in this chapter. With the supplementary information added, the primary link cost of every link is calculated to find the shortest primary path. Then, based on the primary path, backup link cost of every link is decided to find the shortest backup path that optimizes the total network resources.

5.2 Motivation

Three types of supplementary information, primary bandwidth change, backup bandwidth change, and the distribution of traffic for every source and destination pair, are considered in this chapter. The motivation of using these information is described in the following paragraphs.

The first link state information proposed is the primary bandwidth change on every links in MPLS networks. Two traffic statistics, the average holding time of LSPs and the average arrival rate of demands, can be used to compute the amount of primary bandwidth change. By estimating the average holding time of LSPs on a link, we can know the amount of LSPs that will remain on the link at certain time instant. For primary path, when there are more than one widest shortest path available, estimated amount of primary bandwidths which are leaving in the near future become a good metric in making path selection. In this situation, a path that has more bandwidth leaving is a better choice as it helps to reduce congestion in the near future. On the other hand, a path that is crowded with primary LSPs that will last longer is not an optimal choice as it may block requests arriving in the near future.

In addition to that, as primary bandwidth cannot be shared among primary connections, the arrival and departure of a primary LSP normally result in changes that are more significant to the amount of residual free bandwidth than backup LSPs. Therefore, besides estimating the amount of primary bandwidth leaving in duration Δt, the primary LSP arrival rate in the past Δt should be considered to estimate the primary bandwidth that may be required in the next Δt. By considering all these information, a better primary path can be selected to allow more future connections.

For backup path, by estimating the amount of backup bandwidth that is leaving soon on a link, backup bandwidth sharing in network can be improved. If more than one path has identical backup cost, by taking into consideration the backup bandwidth that may be freed in the near future, we can find a second backup cost $c2_{uv}$ (the backup path cost after certain duration) for link(*u,v*). If $c2_{uv}$ is larger than the current backup cost, it means some other backup connections on the links are leaving. As a result, the backup bandwidth is shared by less backup connections. This thus reduces backup bandwidth sharing efficiency.

The traffic distribution of source and destination pair (s-d pair) is suggested to be used to help making routing decision. This information needs not be distributed. Each ingress router needs to know only the traffic distribution of s-d pairs go through itself. By using this information together with the partial information available, an ingress router can estimate the importance of links for the particular s-d pair; thus helps to reduce congestion in the network. For example, when a LSP request arrives at ingress router *s*, *s* will try to find a primary path to destination *d*. If there is a moderately loaded link, where the percentage of bandwidth used by that s-d pair on the link is small, then it is possible that the link is more important to other s-d pair. Therefore, this request should avoid using the link if other path is available. In this way, the supplementary information helps to balance traffic in networks and reduce congestion on links that are critical to some s-d pairs.

5.3 Problem Definition

Consider a MPLS network with *N* nodes and *E* links. The *N* nodes include ingress routers (ingress LER), egress routers (egress LER) and core (transit) routers (LSRs). All the links in the network are unidirectional. LSP setup requests arrive one by one to the ingress LERs. Every LSP terminates after certain duration. Routing decision to set up both the primary and the backup paths is made at the ingress LER where the request arrives. Signaling mechanism such as LDP or RSVP is used to do the actual path set up.

A LSP setup request (*s,d,b*) is defined by its source (*s*), destination (*d*), and the amount of bandwidth required (*b*). Only LSP requests that require protection are considered. For each LSP setup request to be accepted, a primary path and a backup path have to be set up. The primary path should be link disjoint with the backup path to protect the LSP from single link failure. Only single link failure is considered here.

If a request (*s,d,b*) is successfully accepted, *b* units of bandwidth will be reserved on all its primary path links. In order to provide path protection to this LSP setup, backup bandwidth should be reserved when primary path is set up. Backup bandwidth sharing is allowed to minimize the redundancy caused by backup protection. The amount of backup bandwidth sharing that can be achieved on a link depends on the amount of information known about the routing of LSPs currently in progress in the network [4].

The supplementary information includes the primary bandwidth change, backup bandwidth change and the distribution of traffic for every source and destination pair. Link state information, which includes partial information and supplementary information, is disseminated using traffic engineering extensions.

By considering the primary bandwidth change and backup bandwidth change, we can estimate the amount of bandwidth available on the network links in the near future. With this information, better routing and sharing efficiency can be achieved. In addition to that, it is proposed that each ingress node should monitor the distribution of traffic for every source and destination pair that originates from the node. This helps ingress node to make routing decision especially to avoid links, which are relatively more important to other source and destination pair.

A fast heuristic algorithm is proposed to use the supplementary information presented. By using these additional information, the traffic in a network can be better balanced at any instant of time and network resources can be utilized in a more efficient way to accommodate more future requests.

5.4 Supplementary Information

To enable backup path bandwidth sharing, certain link state information needs to be distributed in the network, e.g. total bandwidth used by primary LSPs, total bandwidth used by backup LSPs and residual free bandwidth. These three bandwidth information form the partial information described in [4]. In order to improve primary paths routing and backup bandwidth sharing, additional information (the primary bandwidth change and backup bandwidth change derived after considering LSPs holding time and arrival rate, the distribution of traffic for every source and destination pair) has to be used.

As distributed control is considered, link state information forwarded in the networks should be terse enough to reduce control traffic load. The job of finding the amount of changes in primary and backup bandwidth based on previous primary and backup LSPs average holding time and arrival rate on a link is dispensed to all nodes. Every node, besides forwarding the partial information, needs to find out the primary bandwidth change and backup bandwidth change in the next time duration Δt. The procedure of finding the bandwidth changes is as described in the following paragraphs.

5.4.1 Primary Bandwidth Change

To identify primary path that can reduce congestion in the near future, we need to identify the possible increase and decrease of bandwidth on link(*i,j*) in a short duration Δt. These two quantities can be obtained from past traffic statistic on the link. In order to find the amount of primary bandwidth that will be freed in Δt, first, the average holding time (θ) of *x* previously ended primary LSPs on link(*i,j*) must be calculated. Then, for every current primary LSPs whose (start time + Δt − current time) $\geq \theta$, their bandwidth will be summed up to find the total estimated leaving primary bandwidth α_{ij}.

In addition to the total primary bandwidth leaving in Δt, we need to find the possible amount of primary requests arriving too. To find the total amount of traffic that may arrive in the next duration Δt, traffic arrival rate observed in past duration Δd have to be recorded. This arrival rate is used to find the possible amount of primary bandwidth requested γ_{ij} in next Δt.

Finally, the possible primary bandwidth change is

$$\delta_{ij} = \alpha_{ij} - \gamma_{ij} \tag{5.1}$$

If $\delta_{ij} \geq 0$, it means more free bandwidth may be available on link(*i,j*). If $\delta_{ij} < 0$, the link may become more congested later.

5.4.2 Backup Bandwidth Change

The process of finding backup bandwidth change is identical to the one of finding primary bandwidth change, except that we do not consider possible amount of backup bandwidth that may be requested in the next duration Δt. This is because the amount of backup bandwidth required on a backup path link cannot be calculated explicitly as in the primary bandwidth case.

As backup bandwidth sharing is applied, backup bandwidth used on a link depends on the primary path selected. Also, the amount of backup bandwidth added may be small compared to primary bandwidth (backup bandwidth reserved $\leq$ total backup LSPs bandwidth routed on the link). As a result, for link(*i,j*), only backup bandwidth possibly going to be freed (η_{ij}) will be considered.

In order to find backup bandwidth change on a link, first, the average holding time (θ') of x previously ended backup LSPs on link(*i,j*) must be found. Then, for every current backup LSPs whose (start time + Δt − current time) $\geq \theta'$, their bandwidth will be summed up to find the total estimated leaving backup bandwidth η_{ij} on link(*i,j*).

5.4.3 Source and Destination Pair Traffic Distribution Information

Assume that there are a set of potential source and destination pair denoted by (*g,h*), where *g* is the source node and *h* is the destination node. At every ingress node *g*, for every source and destination pair (*g,h*) originates from that ingress node, a $N \times N$ traffic distribution matrix need to be maintained for both primary paths and backup paths, *N* is the amount of nodes in the network. An entry at row *i*, column *j* corresponds to the amount of primary (backup) bandwidth of a (*g,h*) pair routed on link (*i,j*).

Every time after a LSP request is accepted, both the primary and backup traffic distribution matrices indexed by the request's source and destination pair will be updated by the amount of requested bandwidth according to the primary and backup paths selected. This source and destination pair traffic distribution information needs not be disseminated to other ingress LERs which do routing decisions. Every ingress LER just needs to maintain the information of source and destination pair local to itself.

5.5 Algorithm Description

Active path first heuristic is applied. That is, primary path is searched first and only after a primary path is found, the backup path can be decided. Dijkstra's algorithm is used to find feasible shortest primary path.

For a request (*s,d,b*) that arrives, to start finding a primary path, first, every link that has residual free bandwidth larger or equal to *b* is assigned a link cost equal to *b*. However, for link that is moderately loaded (e.g. more than 50% of the capacity is used), if the percentage of bandwidth used by this s-d pair on the link is small (e.g. less than 50% of the used bandwidth), an extra cost will be added to this link. The extra cost should be large enough so that this link will be used only when there is no other better path available. Dijkstra's algorithm is then used to find the primary path with the minimum cost. If there are many identical minimum cost available, the path with the largest bottleneck link free bandwidth will be selected. Under the situation when there is more than one widest shortest path, the path with the most primary and backup bandwidth leaving soon (in Δt) will be chosen. That is to find the path with maximum

$$\sum_{(i,j)} \delta_{ij} + \eta_{ij}$$

for every link(*i,j*) on the path.

After a primary path is found, backup cost of links must be calculated to find a backup path. A backup path link(*u,v*) has its cost depends on the primary path found.

$$c_{uv} = \begin{cases} 0 & \text{if } M + b \leq B_{uv} \\ min(M + b - B_{uv}, b) & \text{if } M + b > B_{uv} \text{ and } F_{uv} \geq M + b - B_{uv} \\ \infty & \text{Otherwise} \end{cases} \tag{5.2}$$

where

$$M = \max_{(i,j)} P_{ij} \tag{5.3}$$

for every link(*i,j*) selected on primary path. Total primary bandwidth used on a link(*i,j*) is denoted by P_{ij}, total backup bandwidth used on a link(*i,j*) is denoted by B_{ij}, and total free residual bandwidth on link(*i,j*) is denoted by F_{ij}. If $M + b \leq B_{uv}$, it means any link failing on the primary path require at most $M + b$ unit of bandwidth on the links of the backup path. Therefore, no additional bandwidth needs to be reserved on link(*u,v*) as the backup bandwidth B_{uv} on link(*u,v*) is enough to protect the primary path. If $M + b > B_{uv}$, then only B_{uv} unit of bandwidth are shareable, additional reservation of bandwidth ($min(M + b - B_{uv}, b)$) need to be made. $min(M + b - B_{uv}, b)$ tells that the maximum amount of additional bandwidth needed is b. If sufficient free bandwidth cannot be found, then link(*u,v*) is not feasible.

Dijsktra's algorithm is used to find the backup path with minimum cost c_{uv}. However, when multiple identical cost backup paths are found, second backup cost $c2_{uv}$ must be used to select a path.

$$c2_{uv} = \begin{cases} 0 & \text{if } M' + b \leq B_{uv} - \eta_{uv} \\ min(M' + b - B_{uv} + \eta_{uv}, b) & \text{if } M' + b > B_{uv} - \eta_{uv} \\ & \text{and } F_{uv} + \delta_{uv} \geq M' + b - B_{uv} \\ \infty & \text{Otherwise} \end{cases} \tag{5.4}$$

where

$$M' = \max_{(i,j)}(P_{ij} - \delta_{ij}) \tag{5.5}$$

δ_{ij} is the primary bandwidth change on link(*i,j*) in Δ*t* and η_{ij} is the backup bandwidth change on link(*i,j*) in Δ*t*. $c2_{uv}$ considers δ_{ij} and η_{ij} to calculate the link cost on potential backup path link(*u,v*) in Δ*t*. M' is different from M as M' represents the maximum value of primary bandwidth reserved on some link(*i,j*) on the primary path in the near future Δ*t*. For a potential backup path link(*u,v*), if $M' + b \leq B_{uv} - \eta_{uv}$, it means in the near future Δ*t* any link failing on the primary path require at most $M' + b$ unit of bandwidth on the links of the backup path. Therefore, no additional bandwidth needs to be reserved on link(*u,v*) in Δ*t* as the backup bandwidth $B_{uv} - \eta_{uv}$ on link(*u,v*) is enough to protect the primary path. If $M' + b > B_{uv} - \eta_{uv}$, then only $B_{uv} - \eta_{uv}$ unit of bandwidth are shareable in Δ*t*, additional reservation of bandwidth ($min(M' + b - B_{uv} + \eta_{uv}, b)$) need to be made. The path with the minimum $c2_{uv}$ total cost is selected as backup path.

5.6 Performance Study

The simulation results of the scheme proposed are presented in this section. The performance of protection scheme with supplementary information (PSI) is compared with the active path first scheme that uses min-hop algorithm (MHA) and widest shortest path algorithm (WSP). Three performance metrics: the blocking probability, backup bandwidth sharing efficiency and average primary path and backup path length, are used to evaluate the performance of the proposed protection with supplementary information scheme (PSI). An event-driven simulator written in C was used to generate the results.

Three networks are used in the experiments (most simulations are performed in Network 1):

1. Network 1: 18 nodes and 60 links typical ISP network taken from [27] (shown in Figure 5·1).

2. Network 2: 14 nodes and 40 links (shown in Figure 5·2).

3. Network 3: 20 nodes and 64 links (shown in Figure 5·3).

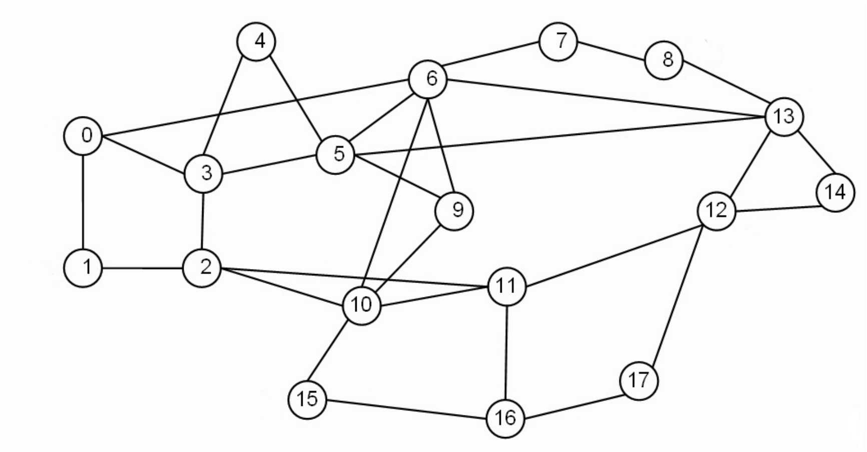

Figure 5·1: Network 1

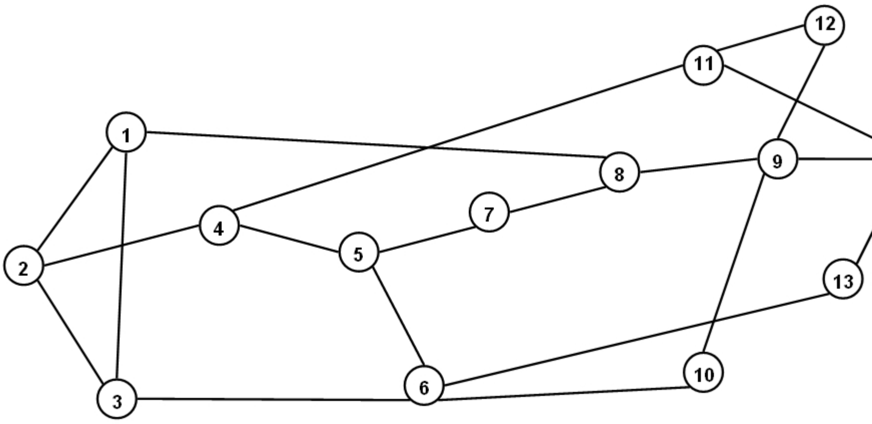

Figure 5·2: Network 2

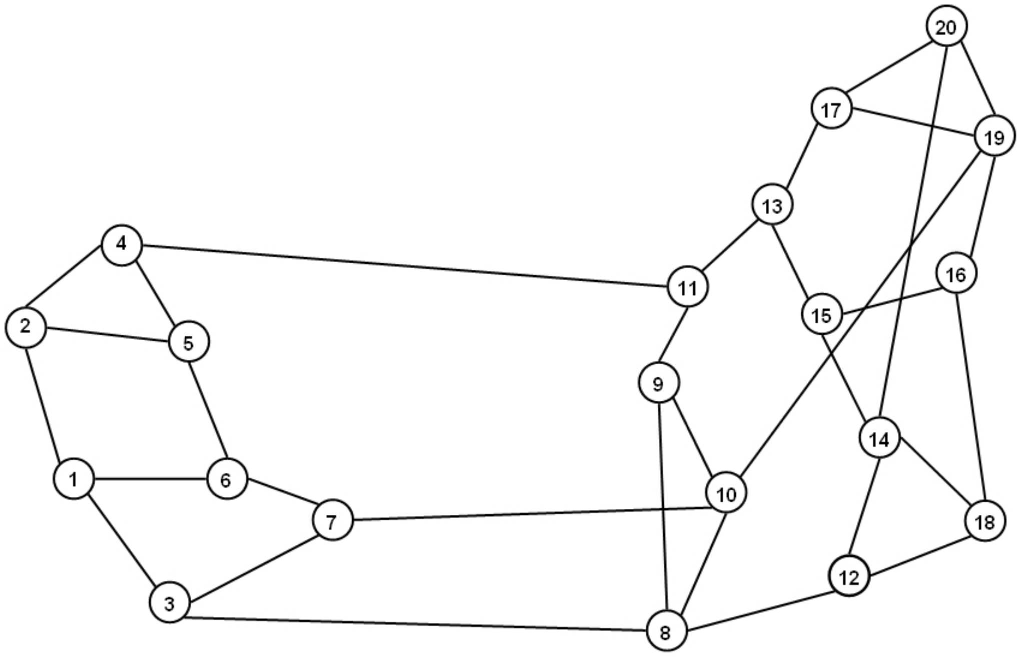

Figure 5·3: Network 3

The capacity of all links in the three networks is 20 units. The LSP setup requests arrive between each source and destination pair according to a Poisson process with average arrival rate λ and the mean exponentially distributed holding times are equal to $1/\mu$. The bandwidths required by LSP setup requests are uniformly distributed between 1 and 3 units. 20000 LSP setup requests are generated in every simulation experiment and every experiment is repeated for 10 different traffic patterns (10 different random seeds).

By varying the traffic load offered to the networks, we can observe that the network performance under low load, moderate load and high load scenarios. Traffic can be controlled by modifying the value of λ/μ. Figures 5·4, 5·5 and 5·6 show the blocking probability of PSI under low load, moderate load and high load scenarios in Network 1. It can be observed that PSI performs well when compare with the MHA and WSP algorithms.

In Figure 5·7, the curves show that the performance of using PSI improves as traffic load increases. Figure 5·8 and 5·9 demonstrate that PSI performs well in Network 2 and Network 3.

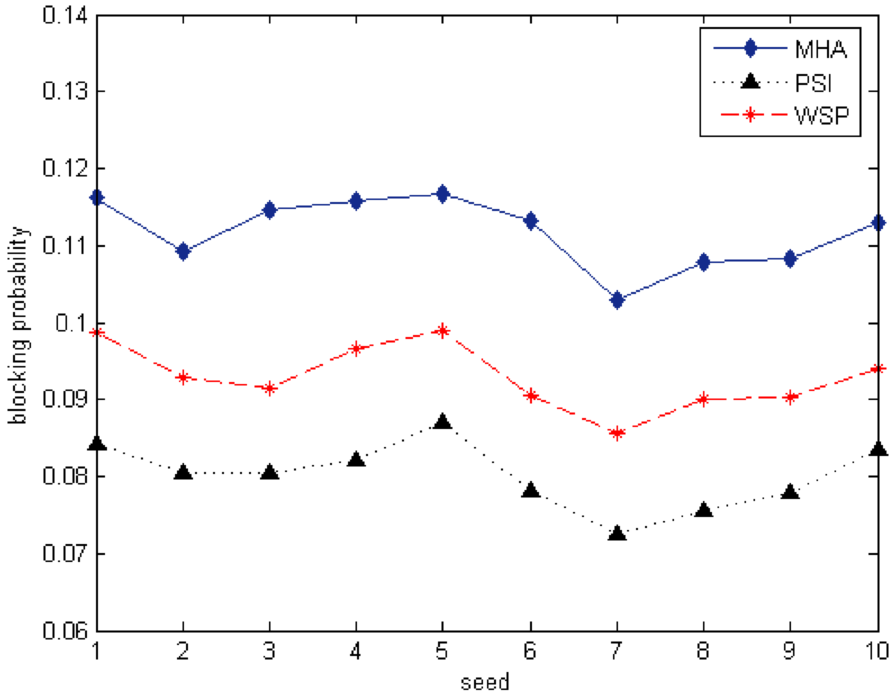

Figure 5·4: Network 1 - Blocking probability under low load scenario

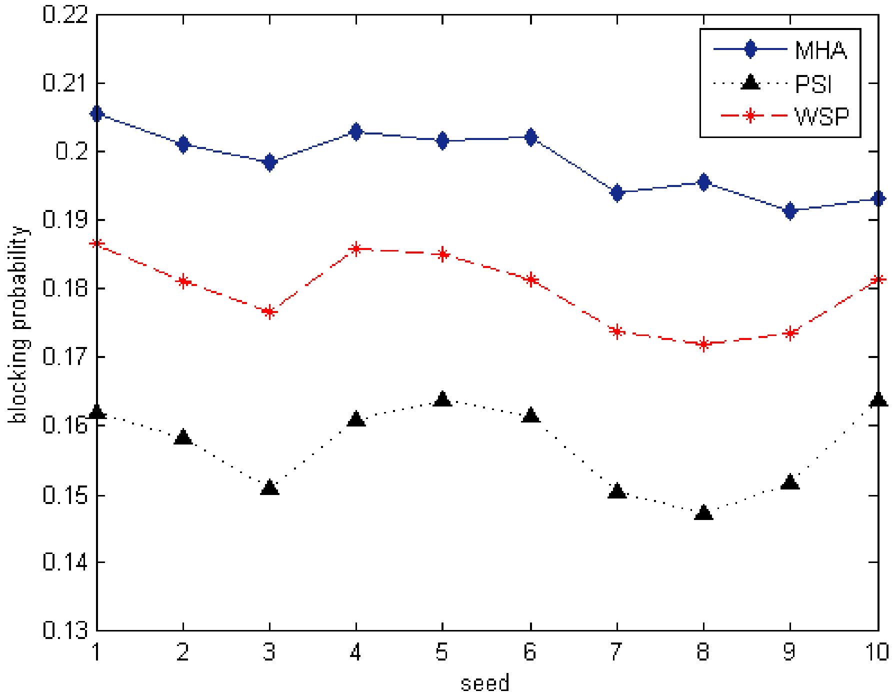

Figure 5·5: Network 1 - Blocking probability under moderate load scenario

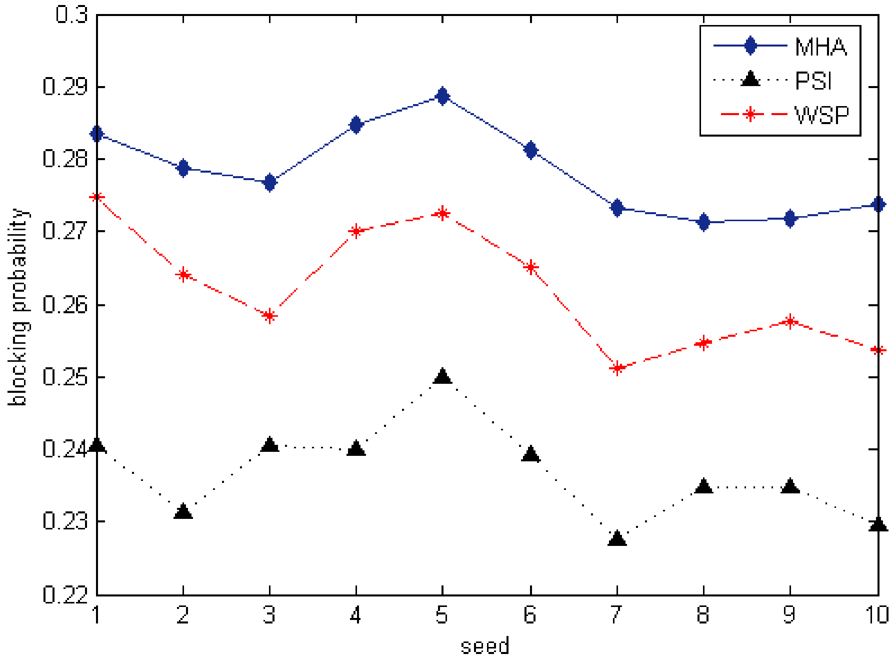

Figure 5·6: Network 1 - Blocking probability under high load scenario

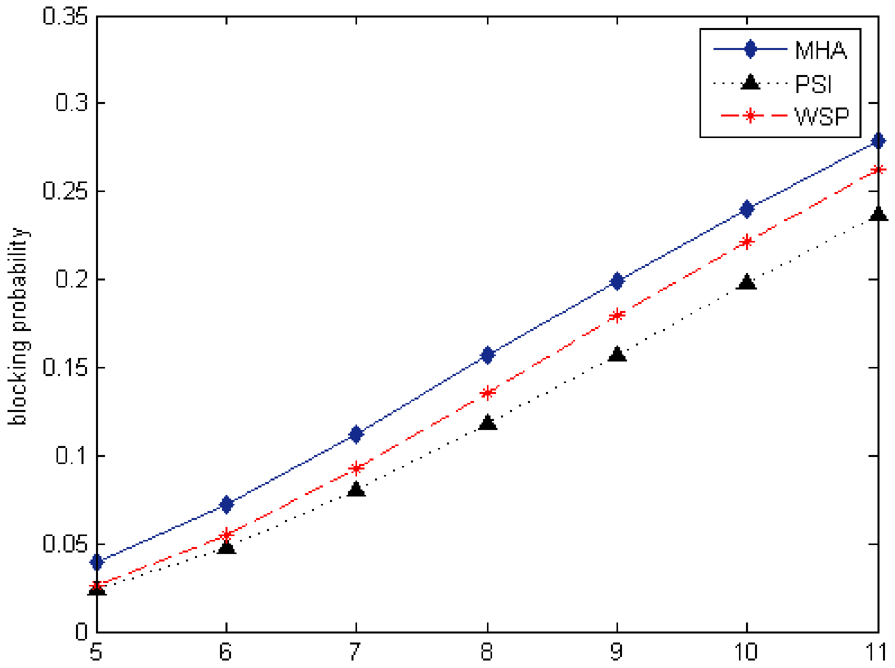

Figure 5·7: Network 1 - Blocking probability vs. load

In addition to the blocking probability, another performance metric used is the backup bandwidth sharing efficiency. Backup bandwidth sharing efficiency is the percentage of backup bandwidth that can be saved with respect to the amount of backup bandwidth used when no backup bandwidth sharing can be achieved. Figure 5·10 shows the backup bandwidth sharing efficiency of PSI in Network 1 under high load scenario. The backup bandwidth required by PSI is much less than the backup bandwidth required by the other two algorithms.

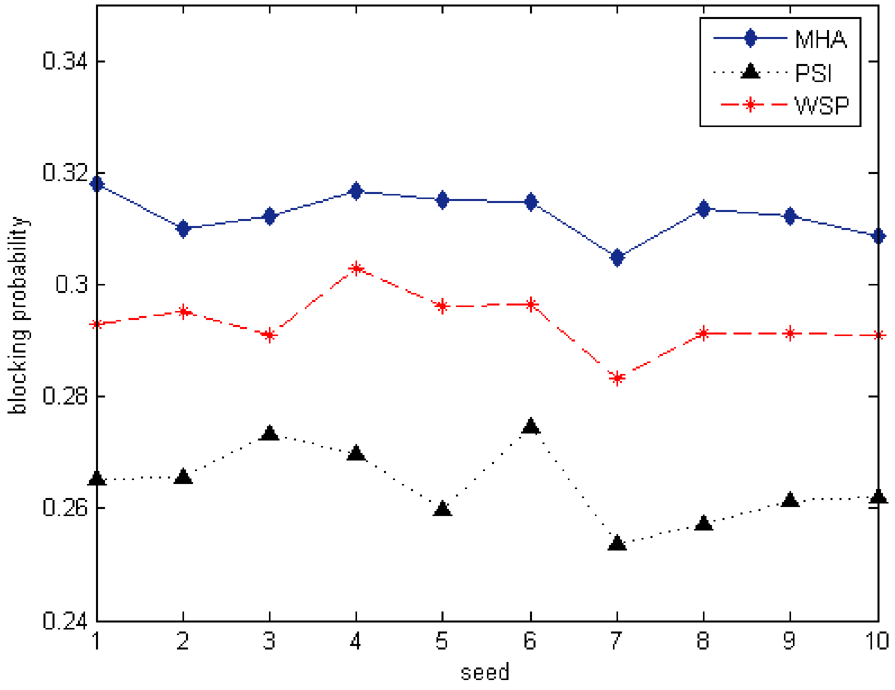

Figure 5·8: Network 2 - Blocking probability under high load scenario

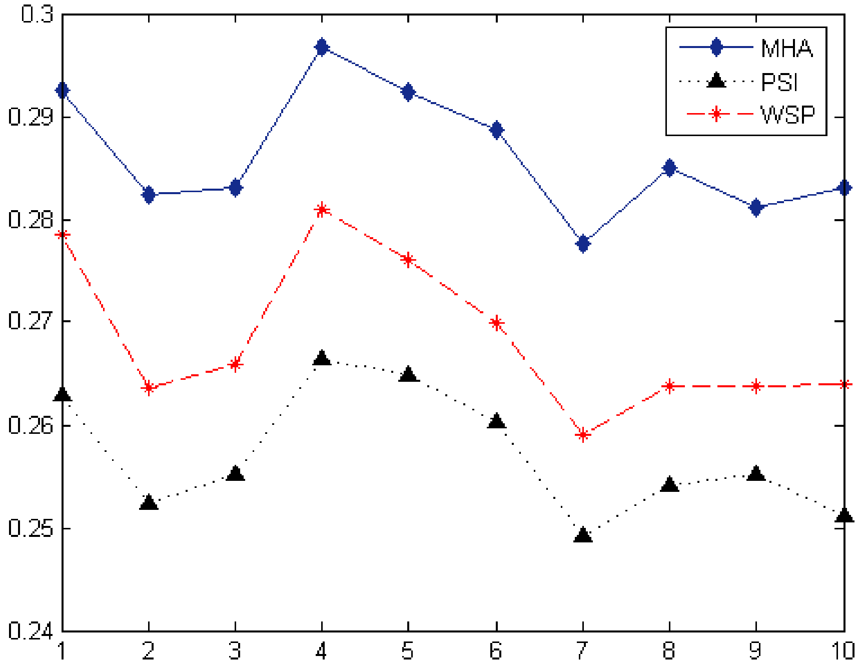

Figure 5·9: Network 3 - Blocking probability under high load scenario

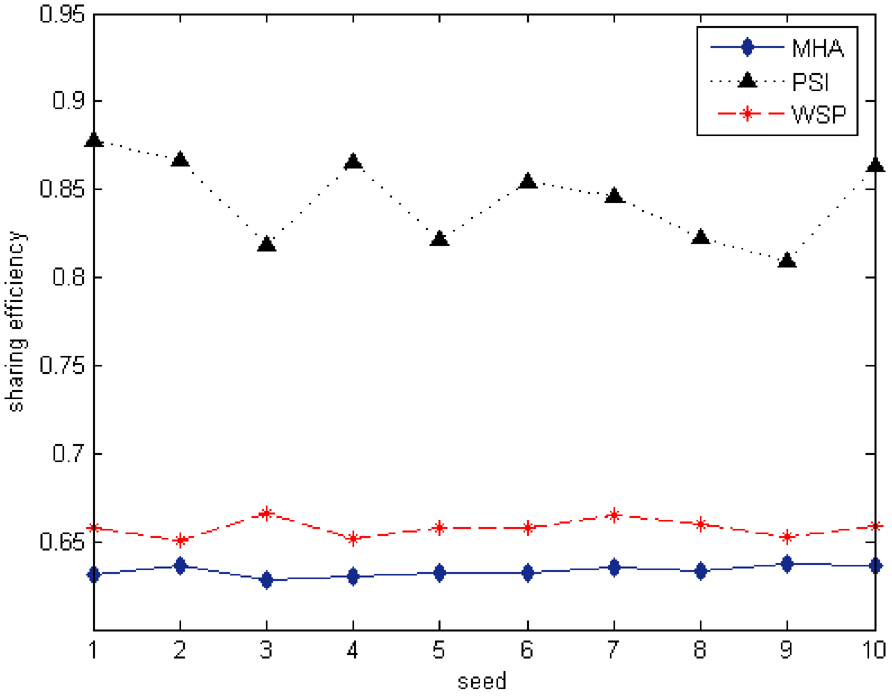

Figure 5·10: Network 1 - Backup bandwidth sharing efficiency under high load

Average primary path and backup path lengths of LSPs accepted in Network 1 using different protection schemes are shown in Figure 5·11. With less than 2% increase in the primary path length, backup path length can be reduced by more than 13% when PSI is used. Accordingly, from Figure 5·12, we can see that the total average primary and backup paths length is shorter when PSI is used. This is because when PSI is used, the backup bandwidth sharing efficiency is higher, as shown in Figure 5·10; as a result, more primary paths and backup paths can be routed on shorter paths.

Extensive simulation experiments conducted above verify the effectiveness of the heuristic proposed. The results show that the heuristic with additional information proposed performs very well though using only shortest path computations.

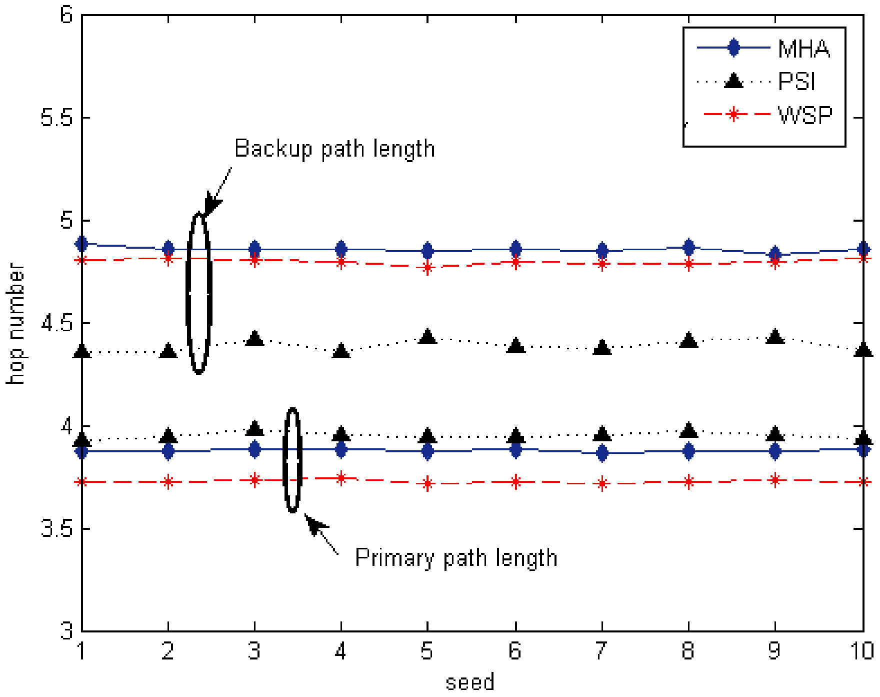

Figure 5·11: Network 1 - Primary path and backup path lengths under high load

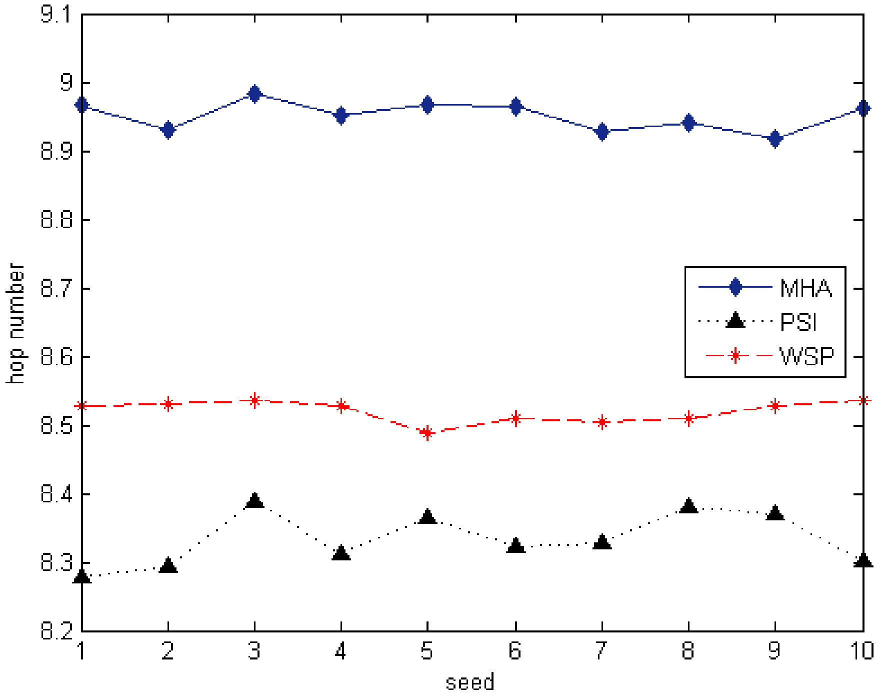

Figure 5·12: Network 1 - Total primary and backup paths length under high load

5.7 Summary

In this chapter, supplementary information that can be used in path protection is introduced to improve the backup bandwidth sharing and traffic load balancing in fault-tolerant MPLS networks. The supplementary information proposed, which consists of primary bandwidth change, backup bandwidth change, and the distribution of traffic for every source and destination pair, together with the partial information described in [4], can be easily disseminated in networks using traffic engineering extensions protocols. Therefore, the information is feasible for distributed networks.

The protection with supplementary information (PSI) heuristic algorithm proposed is an efficient algorithm as it uses only shortest path computations. By considering the supplementary information proposed, the algorithm can reduce LSP setup request blocking probability by increasing backup bandwidth sharing. Besides that, the supplementary information helps to identify critical links for source and destination pairs; thus, helps to balance traffic load in the networks. The effectiveness of the proposed protection with supplementary information scheme has been verified through extensive simulations.

CHAPTER 6

Survivability of Scheduled LSP Demands in MPLS Networks

In this chapter, a path protection algorithm for scheduled LSP demands in MPLS networks is presented. In a scheduled LSP demand, in addition to the source, destination, and the amount of bandwidth required, the connection's setup and teardown times are known. For every demand accepted, a primary path and a backup path are set up at the same time. These two paths should be link disjoint to provide failure-independent path protection. Backup bandwidth sharing is used to reduce excessive resource usage. In this chapter, integer linear programming (ILP) formulations for shared scheduled path protection scheme under single link failure for scheduled LSP demands are developed. As ILP is not desirable for routing of demands in large networks, a heuristic algorithm using shortest path computation is proposed. Extensive simulation experiments are conducted to verify the effectiveness of the proposed heuristic algorithm. The experiments show that the performance of the proposed heuristic algorithm is close to the performance of the ILP.

6.1 Introduction

As current applications require high bandwidth and reliable connections, connection-oriented explicit path routing becomes useful to solve the traffic engineering problem in communication networks. MPLS technique, which uses pre-determined label switched path (LSP), can be used to set up connections for virtual private networks (VPN) and for other applications which use explicit paths. On the other hand, it is found in [39] that identical traffic pattern appeared periodically in the network link observed. This shows that traffic load in a network is predictable. In fact, some of these traffic loads may need to be scheduled at specific times (e.g. certain amount of additional bandwidth may be required by VPN client during office hours).

Traffic demands in networks can be generally classified into several categories based on their characteristics. Static and dynamic traffic models are two of the common traffic models considered in the literatures. In static traffic models, all the demands are known and it is assumed that these demands last forever (i.e., incremental traffic). While in dynamic traffic models, the setup time and holding time of dynamic demands are random. In this chapter, we deal with scheduled traffic demands. Scheduled traffic demands are different from dynamic traffic demands, as the setup time and teardown time of every scheduled demand connection are known in advance. The following example show that bandwidth on a link can be reused by scheduled LSP demands that are not time-overlapped.

In Table 6.1, demand r_1 and demand r_3 are for the same source and destination pair (1, 4). Table 6.1 shows that these two LSP demands are not time-overlapped. As a result, these two LSP demands can take the same path $(1 \rightarrow 2 \rightarrow 5)$, see Figure 6·1,

Table 6.1: Example of four scheduled LSP demands

Demand ID	Source	Destination	Bandwidth	Setup time	Teardown time
r_1	1	4	10	2	14
r_2	3	5	6	5	20
r_3	1	4	3	15	18
r_4	3	5	3	1	4

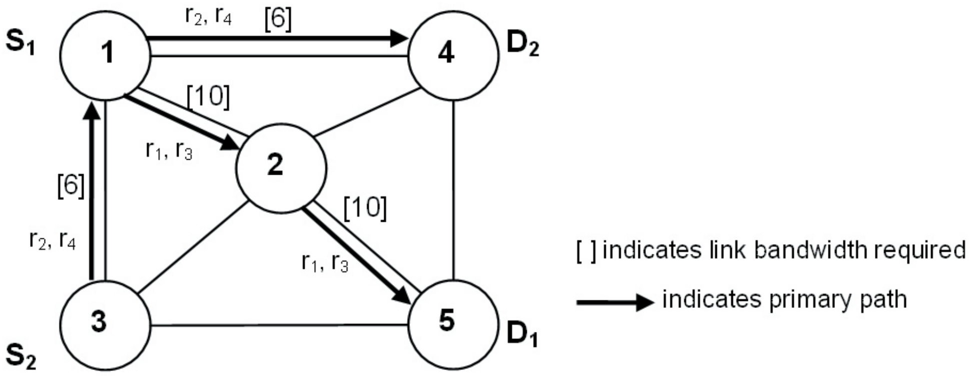

Figure 6·1: Routing of Scheduled LSP Demands

and reduce the amount of bandwidth (only 10 units of bandwidth are used on each link) required by reusing the same bandwidth. The same situation happens to LSP demand r_2 and LSP demand r_4. When shared path protection is implemented, scheduled LSP demands help to achieve saving in the amount of bandwidth required too. Using the same example in Table 6.1 and Figure 6·1, we assume that every scheduled LSP demand must be protected by a link-disjoint backup path and backup bandwidth sharing is allowed to improve bandwidth utilization. In Figure 6·2, it is found that the amount of total bandwidth used in the network is minimized. This is because primary link bandwidth can be reused by LSP demands that are not time-overlapped and backup bandwidth is shared by LSP demands with link-disjoint primary paths.

In this chapter, algorithms to provide path protection for scheduled LSP demands in central-controlled MPLS networks are proposed. By using protection scheme, extra

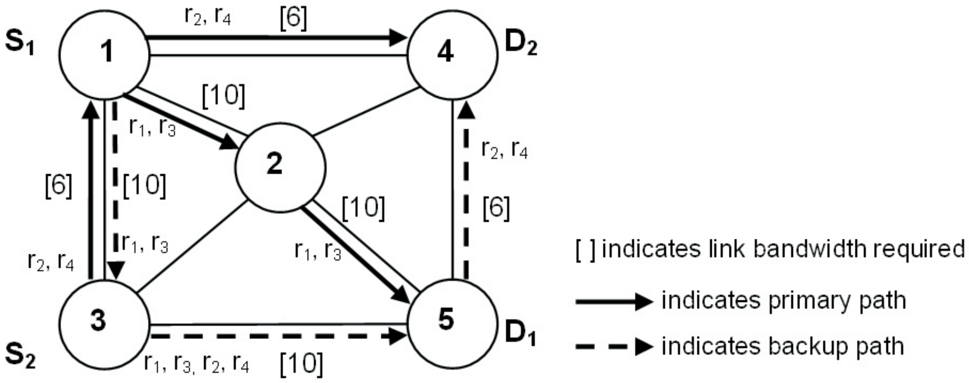

Figure 6·2: Routing of Scheduled LSP Demands With Path Protection

resources must be reserved when setting a connection. Therefore, shared backup bandwidth method is used together with scheduled LSP demands to reduce the total amount of backup bandwidth reserved.

6.2 Related Work

Scheduled lightpath demands are connection demands for which the setup and teardown times are known in advance [21]. In [21], algorithms that compute the routing and wavelength assignment (RWA) for scheduled lightpath demands in a wavelength-switching mesh network without wavelength conversion functionality were presented. The routing problem and the wavelength assignment problem are formulated separately as spatio-temporal combinatorial optimization problems. For the routing problem, a branch and bound algorithm was proposed for exact solution and an alternative tabu search algorithm was proposed for approximate solution. A generalized graph colouring approach is used to solve the wavelength assignment problem. It is believed in [21] that after some years, most of the demands will be either static or scheduled, as the traffic load in a transport network is fairly predictable because of its periodic nature.

The major drawback of the branch and bound algorithm is its exponential time complexity. Therefore, a tabu search algorithm is used to find routing solutions in [21]. Tabu search is an iterative meta-heuristic algorithm used for combinatorial optimization problems. It explores the space of solutions until either a number of iterations is reached or a specific cost criterion is satisfied [21].

In [22], integer linear programming formulations of scheduled demands was developed for both dedicated and shared path protection in wavelength division multiplexing (WDM) networks. There are two objective functions for the ILP problems formulated:

1. minimize the total capacity required for a given traffic demand while providing 100% protection for all connections.

2. given a certain capacity, maximize the number of demands accepted while providing 100% protection for accepted connections.

In MPLS networks, the paths of traffic flows can be given explicitly and can be reconfigured without the interruption of traffic [40]. The method of reconfiguration according to the daily and/or weekly traffic changes is called capacity management [41]. Multi-hour design (MHD) in [42] is a possible approach of capacity management. MHD takes the periodic change of traffic volumes and directions into account by partitioning the whole time scale into several intervals and calculating the maximal traffic demands separately for each interval. On the other hand, a single-hour design (SHD) results in a network that is dimensioned for maximal traffic demands. It is believed in [42] that as the maximal demand between different node pairs may occur in different intervals, the capacities of network devices may be smaller than in the case of single-hour design, resulting in lower deployment cost. The paper [42] proposed an algorithm that is based on the algorithm for single-hour design presented in [43] for multi-hour design problem.

Unlike scheduled demands, demands used in [42] are described by its source, destination, and required capacity. The setup time and teardown time of each demand are unknown. In multi-hour design, the time scale is divided into several intervals. The traffic volumes and distributions within each time interval are estimated according to the daily and/or weekly traffic changes. Nevertheless, both algorithms using scheduled demands and the multi-hour design algorithm try to improve network resources utilization by exploiting the time-disjointness of demands.

6.3 Problem Definition

To provide reliable connection for scheduled LSP demands, a primary path and a backup path need to be set up so that in case of network failure, affected traffic can be sent through the backup path instantaneously. It is assumed that only single link failure can happen at any given time, primary path should be link disjoint with the backup path so that when failure happens, only one path will be affected. Notations used this chapter are as follows:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, the MPLS network is modeled by a connected directed graph $\mathcal{G}$ where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$. $e_k = (i, j)$, $i \neq j$, $i, j \in \mathcal{N}, 1 \leq k \leq E$.
- $\mathcal{L} = \{l_1, l_2, \ldots, l_\gamma\}$ denotes the set of source and destination node-pairs in graph $\mathcal{G}$. $l_k = (s, d)$, $s \neq d$, $s, d \in \mathcal{N}$, $1 \leq k \leq \gamma$, $\gamma \leq N \times (N - 1)$.

- Scheduled LSP demands set $\mathcal{R}$ in $\mathcal{G} = \{r_1^{m_1}, r_2^{m_2}, \ldots, r_z^{m_z}\}, m_1, m_2, \ldots, m_z \in \mathcal{L}$.

  $r_\alpha^m = (s_\alpha^m, d_\alpha^m, b_\alpha^m, t_{arrival,\alpha}^m, t_{end,\alpha}^m),\ 1 \le \alpha \le z.$

  s_α^m is the source node of traffic demand $r_\alpha^m,\ s_\alpha^m \in \mathcal{N}$.

  d_α^m is the destination node of traffic demand $r_\alpha^m,\ d_\alpha^m \in \mathcal{N}$.

  $(s_\alpha^m, d_\alpha^m) = m\ ,\ m \in \mathcal{L}$.

  b_α^m specifies the amount of bandwidth required by traffic demand r_α^m.

  $t_{arrival,\alpha}^m$ specifies the setup time of traffic demand r_α^m.

  $t_{end,\alpha}^m$ specifies the teardown time of traffic demand r_α^m.

- $\mathcal{P}_m =$ potential primary paths for node-pair $m, m \in \mathcal{L}$.

- $\mathcal{Q}_m =$ potential backup paths for node-pair $m, m \in \mathcal{L}$.

- $\Omega_{\mathcal{R}} = \{\omega_1, \omega_2, \ldots, \omega_\psi\}$ is the set of time-overlapped LSP demands at any instant in $\mathcal{G}$. $\omega_\beta = \left[\tau_1^\beta, \tau_2^\beta, \ldots, \tau_z^\beta\right]$, $1 \le \beta \le \psi$, is a $\{0,1\}, 1 \times z$ matrix, only LSP demands which are time-overlapped with all other LSP demands at this instant has value 1 at its entry. The value of ψ depends on the number of different time-overlapped LSP demands sets at any instant in $\mathcal{G}$.

6.3.1 Problems Formulation in Integer Linear Programming

The problem of optimally routing primary path with protection for scheduled LSP demands can be formulated as an integer linear programming (ILP) problem. Decision variables:

- $X_{ij}^{\alpha,m}(p)$ takes on value of 1 if path *p* is used as primary path for traffic demand r_α^m and link(*i,j*) is on *p*, $p \in \mathcal{P}_m$.

- $Y_{ij}^{\alpha,m}(q)$ takes on value of 1 if path *q* is used as backup path for traffic demand r_α^m and link(*i,j*) is on *q*, *q* $\in \mathcal{Q}_m$.

- $X_{ij}^{\alpha,m}$ takes on value of 1 if link(*i,j*) is used on primary path for demand r_α^m.

- $Y_{ij}^{\alpha,m}$ takes on value of 1 if link(*i,j*) is used on backup path for demand r_α^m.

- θ_{ij}^{β} = total amount of primary and backup bandwidth used by time-overlapped traffic demands specified in ω_β on link(*i,j*), $1 \leq \beta \leq \psi$, $(i,j) \in \mathcal{E}$.

- Θ_{ij} = maximum total amount of primary and backup bandwidth used on link(i,j), $(i,j) \in \mathcal{E}$.

The objective of the ILP formulations is to find a pair of primary path and backup path such that the network resources required are minimized. The optimization problem can be formulated as follows:

Objective:

$$\text{Minimize} \sum_{(i,j)\in\mathcal{E}} (\Theta_{ij}) \tag{6.1}$$

subject to:

$$\sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) - \sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) = 1,\ i = s_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.2}$$

$$\sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) - \sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) = 0,\ i \neq s_\alpha^m, d_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.3}$$

$$\sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) - \sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) = -1,\ i = d_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.4}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) - \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) = 1,\ i = s_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.5}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) - \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) = 0,\ i \neq s_\alpha^m, d_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.6}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) - \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) = -1,\ i = d_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.7}$$

$$\sum_{p\in\mathcal{P}_m} \sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) = 1,\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.8}$$

$$\sum_{q\in\mathcal{Q}_m} \sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) = 1,\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.9}$$

$$X_{ij}^{\alpha,m}(p), Y_{ij}^{\alpha,m}(q), X_{ij}^{\alpha,m}, Y_{ij}^{\alpha,m} \in \{0,1\}, \text{ for } \forall\,(i,j) \in \mathcal{E} \text{ and } \forall\, r_\alpha^m \in \mathcal{R}, 1 \le \alpha \le z, m \in \mathcal{L} \tag{6.10}$$

Equations 6.2, 6.3 and 6.4 give the flow balance for the primary paths. Similarly, equations 6.5, 6.6 and 6.7 give the flow balance for the backup paths. Equations 6.8 and 6.9 ensure that only one primary path and one backup path are selected for every demand.

$$X_{ij}^{\alpha,m} = \sum_{p|(i,j)\in p} X_{ij}^{\alpha,m}(p),\ p \in \mathcal{P}_m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.11}$$

$$Y_{ij}^{\alpha,m} = \sum_{q|(i,j)\in q} Y_{ij}^{\alpha,m}(q),\ q \in \mathcal{Q}_m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.12}$$

Equation 6.11 tells whether link(*i,j*) is used on the primary path of demand r_α^m while equation 6.12 tells whether link(*i,j*) is used on the backup path of demand r_α^m. For every demand r_α^m, the primary and backup paths should be link-disjoint. This can be ensured by using the following constraint:

$$X_{ij}^{\alpha,m} + Y_{ij}^{\alpha,m} \le 1, \text{ for } \forall\,(i,j) \in \mathcal{E} \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.13}$$

As all the scheduled LSP demands can be known in advance, optimal backup bandwidth sharing can be achieved by considering the primary and backup paths of all LSP demands. The value of θ_{ij}^{β} depends on the sum of primary paths' bandwidth on link(*i,j*) and the maximum amount of backup bandwidth needed on link(*i,j*) to protect every other links, only time-overlapped traffic demands specified in ω_β are considered:

$$\theta_{ij}^{\beta} \geq \sum_{g=1}^{z} \left(\tau_g^{\beta} \times b_g \times X_{ij}^{g,m} \right) + \sum_{g=1}^{z} \left(\tau_g^{\beta} \times b_g \times X_{uv}^{g,m} \times Y_{ij}^{g,m} \right), \tag{6.14}$$
$$\text{for}\,\forall\, \omega_\beta \in \Omega_{\mathcal{R}}, 1 \leq \beta \leq \psi,\, \forall\,(i,j)\, \in \mathcal{E},\, \text{and}\,\forall\,(u,v) \in \mathcal{E},\, (u,v) \neq (i,j)$$

$$\Theta_{ij} \geq \theta_{ij}^{\beta},\ \text{for}\,\forall(i,j)\, \in \mathcal{E}\,\text{and}\,\forall\,\omega_\beta \in \Omega_{\mathcal{R}}, 1 \leq \beta \leq \psi \tag{6.15}$$

$$\Theta_{ij}, \theta_{ij}^{\beta} \geq 0,\ \text{for}\ \forall(i,j)\, \in \mathcal{E}\,\text{and}\ 1 \leq \beta \leq \psi \tag{6.16}$$

Equation 6.15 finds the maximum amount of bandwidth used on every link. The optimization problem presented above can be solved using CPLEX.

Besides searching the routing solution that minimizes the overall resources used in a network, the problem of maximizing the total number of scheduled demands accepted in a network can be formulated. A scheduled demand can be accepted successfully in a network only if a pair of primary and backup paths can be set up in the network. Every link(*i,j*) in the network has link capacity C_{ij}. In addition to the decision variables described earlier, the following decision variables are needed:

- $\Gamma_{primary}^{\alpha,m}$ takes on value of 1 if a primary path is set up for demand r_α^m.
- $\Gamma_{backup}^{\alpha,m}$ takes on value of 1 if a backup path is set up for demand r_α^m.
- $\Gamma^{\alpha,m}$ takes on value of 1 if demand r_α^m is accepted.

The optimization problem to maximize the number of LSP demands accepted can be formulated as follows:

Objective:

$$\text{Maximize} \sum_{r_\alpha^m \in \mathcal{R}} \left(\Gamma^{\alpha,m}\right) \tag{6.17}$$

subject to:

$$\sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) - \sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) \leq 1,\ i = s_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.18}$$

$$\sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) - \sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) = 0,\ i \neq s_\alpha^m, d_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.19}$$

$$\sum_{j:(j,i)\in\mathcal{E}} X_{ji}^{\alpha,m}(p) - \sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) \leq 1,\ i = d_\alpha^m, \text{ for } \forall\, p \in \mathcal{P}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.20}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) - \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) \leq 1,\ i = s_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.21}$$

$$\sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) - \sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) = 0,\ i \neq s_\alpha^m, d_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{ and } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.22}$$

$$\sum_{j:(j,i)\in\mathcal{E}} Y_{ji}^{\alpha,m}(q) - \sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) \leq 1,\ i = d_\alpha^m, \text{ for } \forall\, q \in \mathcal{Q}_m, \text{ and} \forall\, r_\alpha^m \in \mathcal{R} \tag{6.23}$$

$$\sum_{p\in\mathcal{P}_m} \sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p) \leq 1,\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.24}$$

$$\sum_{q\in\mathcal{Q}_m} \sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q) \leq 1,\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.25}$$

$$X_{ij}^{\alpha,m}(p), Y_{ij}^{\alpha,m}(q), X_{ij}^{\alpha,m}, Y_{ij}^{\alpha,m} \in \{0,1\}, \text{ for } \forall\,(i,j) \in \mathcal{E} \text{ and} \forall\, r_\alpha^m \in \mathcal{R}, 1 \leq \alpha \leq z, m \in \mathcal{L} \tag{6.26}$$

The value of $\Gamma_{primary}^{\alpha,m}$ can be obtained by checking the values of $X_{ij}^{\alpha,m}(p)$ of every possible primary paths between node-pair *m*, for every link(*i,j*) where $i = s_\alpha^m$. The same method is used to find $\Gamma_{backup}^{\alpha,m}$ using $Y_{ij}^{\alpha,m}(q)$ of possible backup paths.

$$\Gamma_{primary}^{\alpha,m} = \sum_{p\in\mathcal{P}_m} \sum_{j:(i,j)\in\mathcal{E}} X_{ij}^{\alpha,m}(p),\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.27}$$

$$\Gamma_{backup}^{\alpha,m} = \sum_{q\in\mathcal{Q}_m} \sum_{j:(i,j)\in\mathcal{E}} Y_{ij}^{\alpha,m}(q),\ i = s_\alpha^m, \text{ for } \forall\, r_\alpha^m \in \mathcal{R} \tag{6.28}$$

For a LSP setup request to be accepted in MPLS network, a pair of primary and backup paths must be found, else the request will be dropped. The following equation 6.29 makes sure that both primary and backup path are set up at the same time.

$$\Gamma^{\alpha,m}_{primary} = \Gamma^{\alpha,m}_{backup}, \text{ for } \forall\, r^m_\alpha \in \mathcal{R} \tag{6.29}$$

$$\Gamma^{\alpha,m} = \Gamma^{\alpha,m}_{primary}, \text{ for } \forall\, r^m_\alpha \in \mathcal{R} \tag{6.30}$$

$$X^{\alpha,m}_{ij} = \sum_{p|(i,j)\in p} X^{\alpha,m}_{ij}(p),\ p \in \mathcal{P}_m,\ \text{for} \forall\, r^m_\alpha \in \mathcal{R} \tag{6.31}$$

$$Y^{\alpha,m}_{ij} = \sum_{q|(i,j)\in q} Y^{\alpha,m}_{ij}(q),\ q \in \mathcal{Q}_m,\ \text{for} \forall\, r^m_\alpha \in \mathcal{R} \tag{6.32}$$

$$X^{\alpha,m}_{ij} + Y^{\alpha,m}_{ij} \leq 1,\ \text{for} \forall\, (i,j) \in \mathcal{E}\, \text{and} \forall\, r^m_\alpha \in \mathcal{R} \tag{6.33}$$

$$\begin{gathered} \theta^\beta_{ij} \geq \sum_{g=1}^{z} \left(\tau^\beta_g \times b_g \times X^{g,m}_{ij}\right) + \sum_{g=1}^{z} \left(\tau^\beta_g \times b_g \times X^{g,m}_{uv} \times Y^{g,m}_{ij}\right), \\ \text{for } \forall\, \omega_\beta \in \Omega_{\mathcal{R}}, 1 \leq \beta \leq \psi,\ \forall\, (i,j)\ \in \mathcal{E},\ \text{and } \forall\, (u,v) \in \mathcal{E},\ (u,v) \neq (i,j) \end{gathered} \tag{6.34}$$

$$\Theta_{ij} \geq \theta^\beta_{ij},\ \text{for} \forall\, (i,j) \in \mathcal{E}\, \text{and} \forall\, \omega_\beta \in \Omega_{\mathcal{R}},\ 1 \leq \beta \leq \psi \tag{6.35}$$

$$\Theta_{ij} \leq C_{ij},\ \text{for} \forall\, (i,j) \in \mathcal{E} \tag{6.36}$$

$$\Theta_{ij}, \theta^\beta_{ij} \geq 0,\ \text{for } \forall\, (i,j) \in \mathcal{E}\, \text{and}\, 1 \leq \beta \leq \psi \tag{6.37}$$

Equation 6.36 ensures that the bandwidth used by primary connections and backup connections on any link(*i,j*) will not exceed the link capacity C_{ij}.

6.4 Heuristic Algorithm Description

When ILP is used to find the optimal bandwidth allocation for scheduled demand requests, excessive processing time is required even with small number of requests in small network; thus, making ILP infeasible for medium and large size networks. A fast heuristic algorithm is proposed in this section to provide efficient bandwidth allocation for scheduled demand requests while providing reliable protection for every setup connections. This algorithm is good in scalability, therefore, allows the routing and protection for large networks.

The proposed heuristic algorithm aims to utilize the property (the knowledge of setup time and teardown time) of scheduled LSP demands in order to achieve efficient demand routing. An interval graph is used to show the relationship of demands in term of their setup time and teardown time in a network. Consider the same problem formulated in section 6.3, the number of nodes in an interval graph **G** is equal to the number of demands in $\mathcal{R}$. **G** is an undirected and weighted graph. Every node α in **G** represents the demand r_{α}^{m} in $\mathcal{R}$ and is associated with a positive weight $\vartheta_{\alpha} = b_{\alpha}^{m}$, which is the amount of bandwidth requested by demand r_{α}^{m} , $1 \leq \alpha \leq z$ and $m \in \mathcal{L}$. There exists a link $\ell_{\alpha_1 \alpha_2}$ between node α_1 and node α_2 if demand $r_{\alpha_1}^{m_1}$ is time-overlapped with demand $r_{\alpha_2}^{m_2}$, $\alpha_1 \neq \alpha_2, 1 \leq \alpha_1, \alpha_2 \leq z$ and $m_1, m_2 \in \mathcal{L}$.

To explore the time-disjointness of scheduled LSP demands in **G**, cliques need to be found from **G**. A clique is a complete subgraph of a graph. A graph is complete if all of its vertices are pairwise adjacent, that is, every vertex is connected to all other vertices in the graph. The maximum clique problem asks for clique of maximum cardinality (the cardinality of a set *S*, is the number of elements in *S*). The maximum clique problem is a well-known example of intractable combinatorial optimization problem [44] and was proved to be NP-complete in [45]. Due to the computational complexity of finding maximum clique, exact algorithms are guaranteed to return a solution in a time that increases exponentially with the number of vertices in the graph. Therefore, those algorithms are inapplicable even to moderately large problem instances. In regards to that, efficient heuristic has to be used to find maximum clique in a graph.

A clique in **G** represents all the demands that are time-overlapped with each other. Thus, to route all the demands in $\mathcal{R}$, all the cliques in **G** must be found. In order to find the maximum clique, different algorithms have been proposed in [46], [47], [48], [49], [50], [51]. A sequential greedy algorithm [46] which can run very fast is used in the proposed algorithm. This greedy algorithm finds a maximum clique through the repeated addition of a vertex into a partial clique or the repeated deletion of a vertex from a set that is not a clique. Decisions on which vertex to be added in or moved out next are based on certain indicators associated with candidate vertices.

After maximum clique is found from **G**, all time-overlapped demands in the maximum clique should be routed efficiently so that the overall primary bandwidth required is reduced and the total amount of backup bandwidth shared is maximized. A primary path and a backup path must be set up for a LSP setup request to be accepted. In regards to the fact that all the scheduled LSP demands are known in advance, complete routing information can be obtained and thus efficient backup paths sharing can be achieved. A backup path link(*u,v*) has its cost depends on the primary path found. σ_{ij}^{uv} gives the amount of primary bandwidth on link(*i,j*) that is protected on link(*u,v*). The cost of using link(*u,v*) on the backup path if link(*i,j*) is used on the primary path, $\wp_{ij}^{uv}$, is:

$$\wp_{ij}^{uv} = \begin{cases} 0 & \text{if } \sigma_{ij}^{uv} + b \leq B_{uv} \text{ and } \textit{(i,j)}\neq\textit{(u,v)} \\ min(\sigma_{ij}^{uv} + b - B_{uv},\, b) & \text{if } \sigma_{ij}^{uv} + b > B_{uv} \text{ and } F_{uv} \geq \sigma_{ij}^{uv} + b - B_{uv} \\ & \text{and } \textit{(i,j)}\neq\textit{(u,v)} \\ \infty & \text{Otherwise} \end{cases} \tag{6.38}$$

where B_{uv} is the total backup bandwidth used on a link(*u,v*), F_{uv} is the total free residual bandwidth on link(*u,v*), and *b* is the amount of bandwidth required by the LSP setup request. $\wp_{ij}^{uv}$ represents the amount of additional backup bandwidth needed on link(*u,v*) if link(*i,j*) is used in the primary path. If $\sigma_{ij}^{uv} + b \leq B_{uv}$, it means any link failing on the primary path require at most $\sigma_{ij}^{uv} + b$ unit of bandwidth on the links of the backup path. Therefore, no additional bandwidth needs to be reserved on link(*u,v*) as the backup bandwidth B_{uv} on link(*u,v*) is enough to protect the primary path. If $\sigma_{ij}^{uv} + b > B_{uv}$, then only B_{uv} unit of bandwidth are shareable, additional reservation of bandwidth ($min(\sigma_{ij}^{uv} + b - B_{uv}, b)$) need to be made. $min(\sigma_{ij}^{uv} + b - B_{uv}, b)$ tells that the maximum amount of additional bandwidth needed is b. If sufficient free bandwidth cannot be found, then link(*u,v*) is not feasible.

The actual backup link cost ϱ_{uv} of link(*u,v*) is:

$$\varrho_{uv} = \max_{(i,j)} \wp_{ij}^{uv} \tag{6.39}$$

for every link(*i,j*) selected on primary path.

The heuristic algorithm proposed is as follows:

1. Let $\mathbf{G}' = \mathbf{G}$.

2. Find the maximum clique for $\mathbf{G}'$. $\mathbf{G}'$ is a subset of $\mathbf{G}$. $\mathbf{G}'$ is formed by demands that has not been assigned primary and backup paths, and demands that have been assigned paths but have more than one neighbour that has not been assigned paths.

    (a) Select the most connected vertex α_1 in $\mathbf{G}'$. Let $\mathbf{\Delta} = \{\alpha_1\}$, **REST** $= \mathbf{G}'$.

    (b) Let α_2 be the vertex connects to all vertices in $\mathbf{\Delta}$ and is the vertex connected to most other vertices in **REST**, $\alpha_2 \in$ **REST** and $\alpha_2 \notin \mathbf{\Delta}$.

    (c) If α_2 cannot be found, halt and return $\mathbf{\Delta}$.

    (d) Set $\mathbf{\Delta} = \mathbf{\Delta} \cup \{\alpha_2\}$, **REST** = **REST** $\setminus$ { vertices not connected to α_2 }.

    (e) Go to step 2b.

3. Let $\mathcal{G}' = \mathcal{G}$.

4. Update graph $\mathcal{G}'$ with the amount of bandwidth used by demands that have been assigned primary and backup paths in $\mathbf{\Delta}$.

5. Find the unsolved vertex α' with the heaviest weight in $\mathbf{\Delta}$. A vertex α' is unsolved if it represents the demand that has not been assigned primary and backup paths. There can be solved vertex in $\mathbf{\Delta}$ (see step 2). Weight of α' is $\vartheta_{\alpha'}$.

6. Find the minimum hop primary path p_1 that has sufficient free bandwidth in graph $\mathcal{G}'$ for demand $r^m_{\alpha'}$ represented by vertex α' in $\mathbf{\Delta}$, $p_1 \in \mathcal{P}_m$, $m \in \mathcal{L}$. If there is more than one minimum hop path, select the widest path that has the most free bandwidth unused by previous demands.

7. If no primary path can be found, go to step 12.

8. After a primary path p_1 is found, continue to find a least cost backup path p_2, which is link disjoint with the primary path p_1. p_2 can be computed using Dijkstra's algorithm with the link cost ϱ_{uv} defined earlier for every link(*u*,*v*), u, v are vertices in $\mathcal{G}'$.

9. If no backup path can be found, go to step 12.

10. Set up both primary and backup paths for demand $r^m_{\alpha'}$ represented by vertex α'.

11. If all demands connected to α' are solved, remove α' from $\mathbf{G}'$. Go to step 13.

12. Remove α' from $\mathbf{G}'$ and $\mathbf{\Delta}$, as a pair of primary and backup path cannot be found.

13. If all the demands in $\mathbf{\Delta}$ have been assigned paths, and $\mathbf{G}'$ is not empty, go to step 2. Else, continue to route demands in $\mathbf{\Delta}$, go to step 4. If $\mathbf{G}' = \emptyset$, exit as all the scheduled LSP demands have been processed.

6.5 Experimental Results

This section presents the simulation results of the proposed algorithm. The performance of the proposed algorithm using scheduled LSP demands is compared with the performance of network using ILP with scheduled LSP demands, as well as the performance of network using ILP with static demands. The experimental set up is the following. Experiments are performed on three networks:

- Network 1: 15 nodes, 52 links (see Figure 6·3).
- Network 2: 18 nodes, 60 links (see Figure 6·4).
- Network 3: 70 nodes, 264 links.

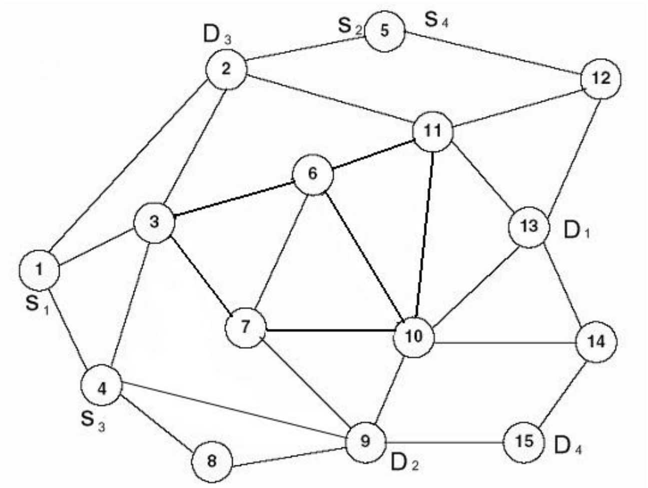

Figure 6·3: Network 1

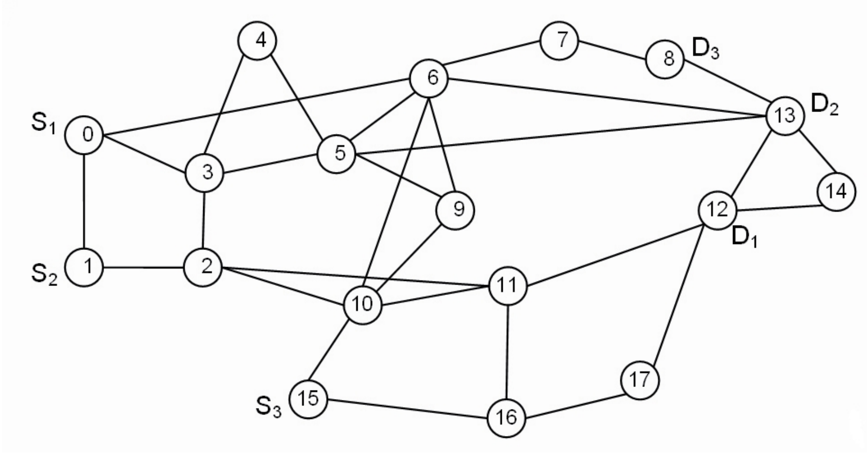

Figure 6·4: Network 2

Each undirected link in Figures 6·3 and 6·4 represents two directed links. We consider two types of traffic, scheduled demands and static demands. The bandwidth required by these scheduled and static LSP demands are uniformly distributed between 1 and 10 units. Each LSP demand must be allocated a primary path and a backup path. Sharing of backup paths bandwidth is allowed.

The following two performance metrics are used to study the performance of different schemes considered.

- Total bandwidth consumed: For the set of experiments to obtain this metric, the capacity of each link is set to infinity. Therefore, all the LSP demands in the network will be satisfied. The objective is to compare the total amount of bandwidth consumed using proposed algorithm with scheduled LSP demands to the total amount of bandwidth consumed using ILP with scheduled demands and with static demands.

- Total number of demands dropped: To find the total number of demands dropped, it is assumed that every link has finite capacity. Every link is restricted to have a capacity of 20 units. Thus, not all demands are satisfied. Experiments with this performance metric aim to present the behaviour of different schemes with respect to the total number of demands dropped.

CPLEX is used to solve the ILP problems for both scheduled and unscheduled LSP demands in the MPLS networks. Computing the paths for all LSP demands using CPLEX can take hours. However, using the proposed algorithm to route scheduled LSP demands is relatively much faster.

Figures 6·5 and 6·6 compare the total amount of bandwidth consumed by the three schemes in Network 1 and Network 2. The saving in total bandwidth consumed between proposed algorithm for scheduled LSP demands and ILP for static demands ranges from about 40% to 50%. The saving in total bandwidth consumed between ILP for scheduled LSP demands and ILP for static demands is between 45% and 60%. The total amount of bandwidth consumed by proposed algorithm for scheduled LSP demands is close to the total amount of bandwidth used by ILP for scheduled LSP demands in both Network 1 and Network 2. This is surprising considering the fact that optimization method using

ILP requires much more time than the heuristic algorithm proposed.

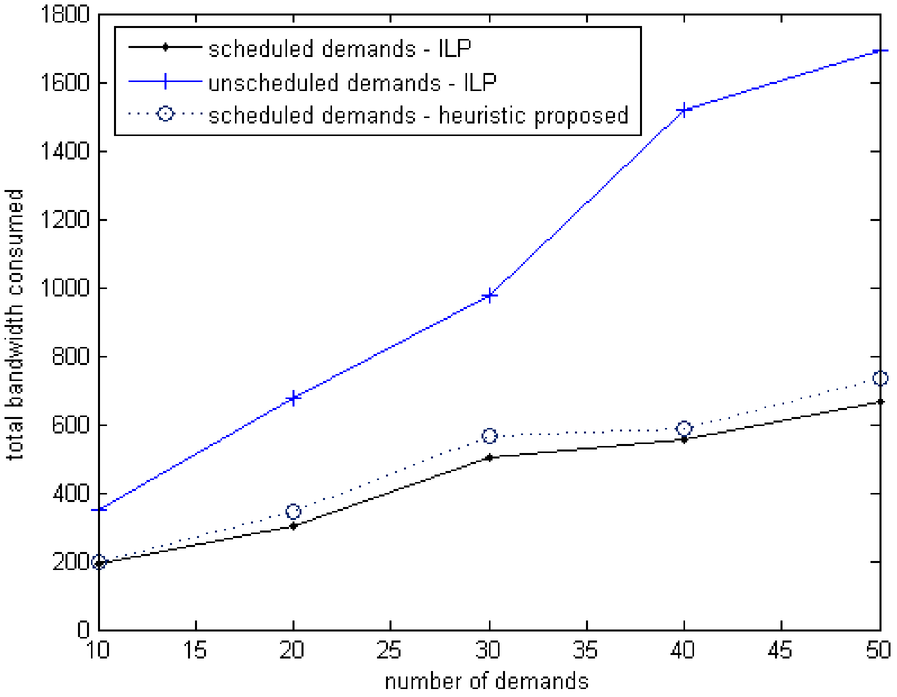

Figure 6·5: Network 1 - Total bandwidth consumed vs. Number of LSP demands

Total amount of bandwidth consumed by scheduled LSP demands in Network 3 is computed to show that the proposed algorithm works well with large networks too. The result is shown in Figure 6·7. It is found that total amount of bandwidth consumed is significantly reduced when the proposed heuristic algorithm is used with scheduled LSP demands.

Another set of experiments were performed to study the behaviour of different schemes with respect to the total number of demands dropped. 10 experiments were performed in Network 1 and Network 2 respectively. 50 demands are loaded into each network. Figure 6·8 shows the number of LSP demands dropped in Network 1. Figure 6·9 gives the number of LSP demands dropped in Network 2. It is found that the proposed heuristic performs considerably better than the ILP using static demands. More importantly,

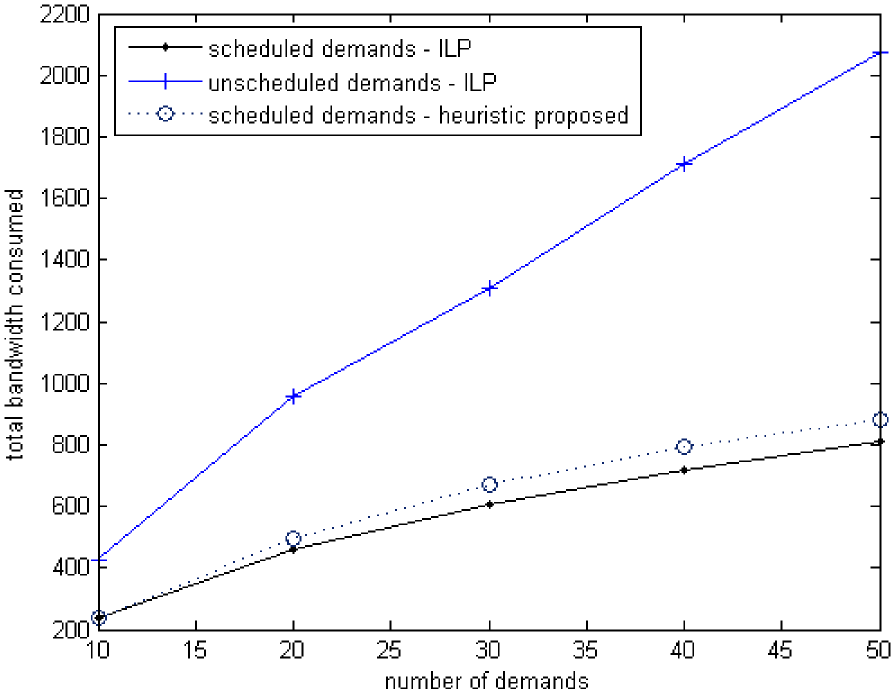

Figure 6·6: Network 2 - Total bandwidth consumed vs. Number of LSP demands

the performance of the proposed heuristic algorithm for scheduled LSP demands is very close to that of the ILP for scheduled demands. This shows that the proposed heuristic algorithm, which is fast and scalable, can be used for effective routing of restorable scheduled LSPs.

Figure 6·10 presents the number of LSP demands dropped in Network 3 when 100 demands are inserted into the network. It is found that the results obtained in Figure 6·10 are consistent with the results found in Figure 6·8 and Figure 6·9. The heuristic algorithm works well in reducing the number of demands dropped in large network.

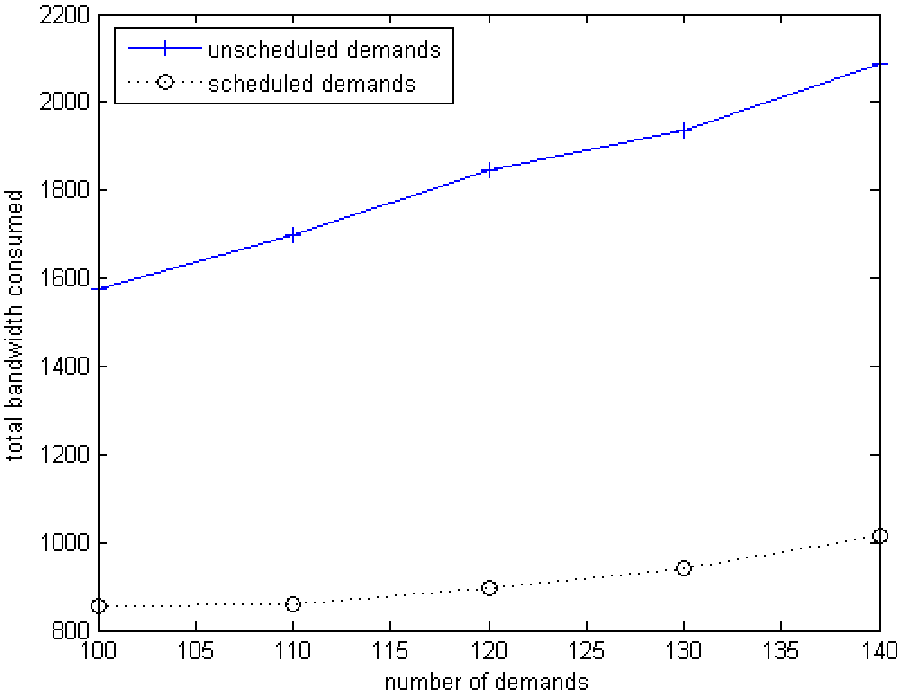

Figure 6·7: Network 3 - Total bandwidth consumed vs. Number of LSP demands

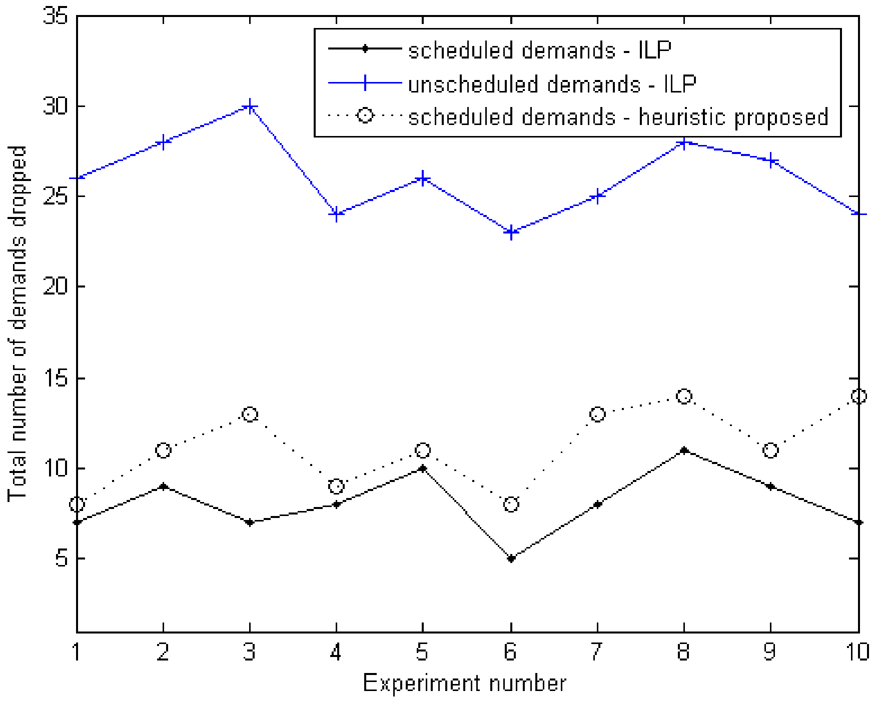

Figure 6·8: Network 1 - Total number of demands dropped for 10 random experiments

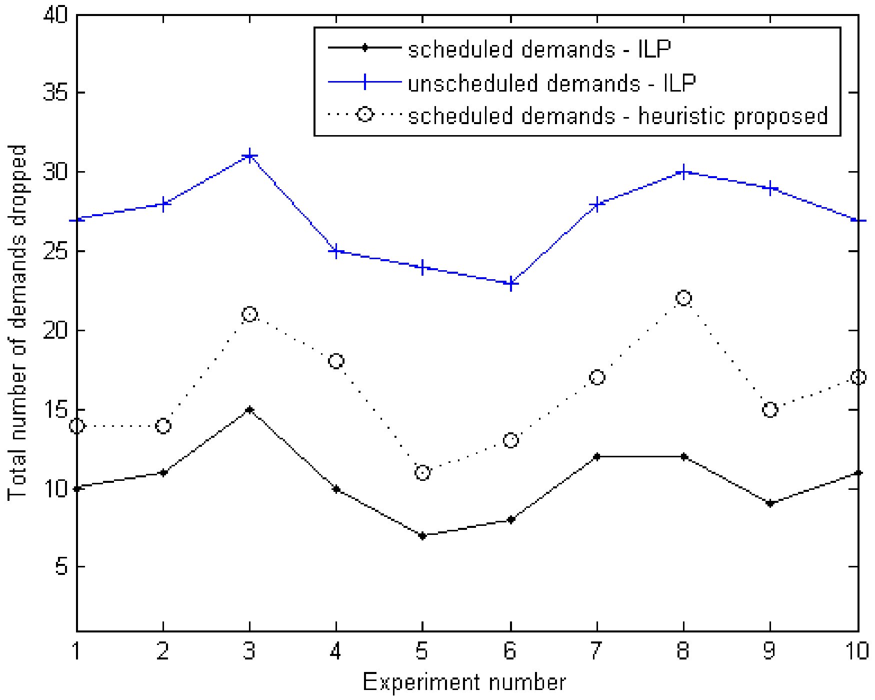

Figure 6·9: Network 2 - Total number of demands dropped for 10 random experiments

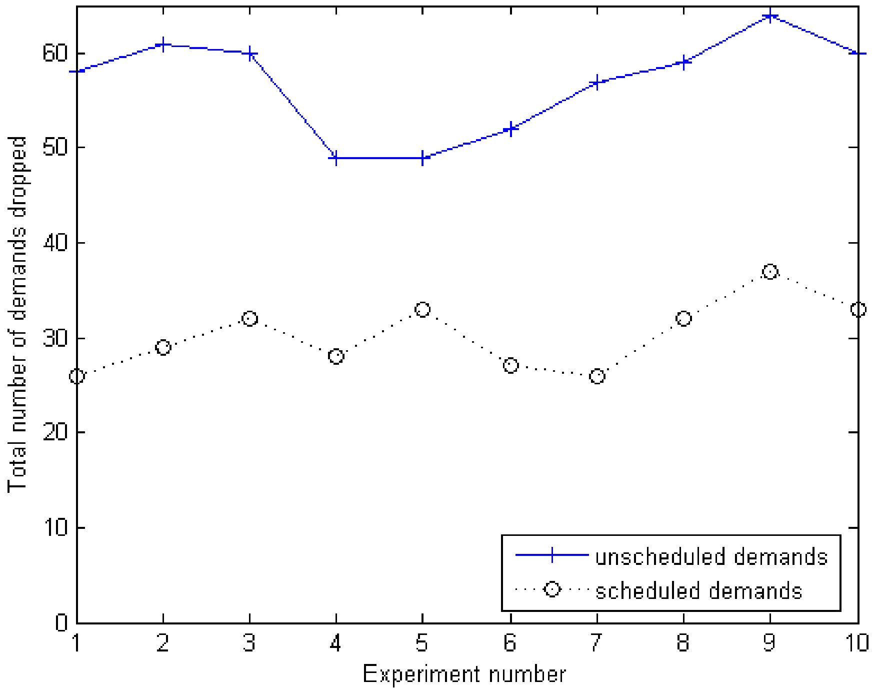

Figure 6·10: Network 3 - Total number of demands dropped for 10 random experiments

6.6 Summary

In this chapter, optimal routing of scheduled LSP demands in MPLS networks is explored. When a scheduled demand is considered, the source and destination together with the setup time and teardown time of the connection are known. These information allow the LSP connections to be better scheduled and thus, reducing link congestion in MPLS networks.

Optimal routing of scheduled LSP demands in a network can be achieved using integer linear programming (ILP). However, ILP is not time efficient and is bad for scalability. A heuristic algorithm utilizing the characteristics of scheduled demand is developed to route scheduled LSP demands fast and efficiently. The proposed heuristic algorithm is good for scalability too.

To optimize scheduled LSP demand connections whose setup time and teardown time are known, the heuristic algorithm must be able to reuse link bandwidths for LSP connections that are time disjoint. At the same time, bandwidth used by time-overlapped demands should be optimized. Backup bandwidth should be minimized by improving the backup bandwidth sharing efficiency.

Interval graph formed by all the scheduled LSP demands known can be used to find out the groups of time-overlapped scheduled LSP demands. Every node in the interval graph is a scheduled demand known. There exists a link between two nodes if the two scheduled LSP demands represented by the nodes are time-overlapped. A group of scheduled LSP demands that are time-overlapped in the interval graph can be identified as a clique in the interval graph. A clique is a complete subgraph of a graph. In order to find the maximum cliques in **G**, a sequential greedy heuristic [46] which can run very fast is used in the proposed heuristic algorithm.

After a maximum clique is found from the interval graph, every demand in the clique is allocated a primary path and a backup path. Algorithm developed to find the pair of path is fast and efficiently reduces bandwidth wastage.

Simulations have been performed to verify that the performance of the proposed heuristic algorithm is close to the performance of ILP for scheduled LSP demands in MPLS networks. It is shown that the proposed algorithm works well in large network too. The proposed heuristic algorithm succeeds to achieve two goals: minimize the total bandwidth used and maximize the number of requests accepted.

CHAPTER 7

Conclusions

This chapter concludes the thesis with a summary of the contributions of the research made. This thesis was aimed to develop fast protection algorithms for traffic with strict time requirements and reliability requirements in MPLS networks. The objectives set out for this thesis have been achieved.

The major contributions are summarized in the following paragraphs.

1. **Multipath protection** - Instead of allocating a spare backup path to protect the primary path, multiple backup paths are used to protect the primary path. Backup bandwidth sharing is allowed to reduce extra bandwidth spent for protection. Two multipath protection algorithms are proposed in this thesis. The advantages of using multiple backup paths include:

   - Reduce link congestion: Traffic load between every source-destination pair is transmitted through multiple paths. This prevents free bandwidth available on critical network links from being exhausted early and becoming a bottleneck.

   - Increase the number of LSP setup requests accepted: A LSP setup request is accepted only when a pair of primary and backup paths can be set up. Therefore, a LSP setup request may be dropped even though a primary path can be found. Multiple backup paths protection reduces the amount of bandwidth required on each backup path; it helps to avoid dropping LSP setup request because a single backup path satisfying the bandwidth requirement cannot be found.

- Improve backup bandwidth sharing: By distributing the backup bandwidth needed on separate link-disjoint paths, backup bandwidth reserved on each backup path is smaller. More backup connections that are link disjoint can share the same backup bandwidth. This improves the backup bandwidth sharing efficiently; thus, reduces the extra spare bandwidth needed to protect every primary connection.

2. **Path protection with supplementary information** - Additional information obtained from network are used to improve the performance of demands routing in distributed MPLS network. The information includes:

   i. The primary bandwidth change on links.

   ii. The backup bandwidth change on links.

   iii. The distribution of traffic for every source and destination pair.

   The average connections holding time and average arrival rate of demands on a link can be used to estimate the primary bandwidth change and backup bandwidth change on the link. This helps to reduce congestion in networks. The distribution of traffic for every source and destination pair helps every ingress LER to identify links that are critical to certain source and destination pairs. Accordingly, it reduces the blocking rate of requests. All the additional information can be disseminate in MPLS using traffic-engineering extensions.

3. **Fast protection for scheduled LSP demands** - The optimal routing of scheduled LSP demands in MPLS networks is explored. Scheduled LSP demands are connection demands for which the source, destination, setup time and teardown time are known in advance. Scheduled LSP demands allow bandwidth on a link to be reused by demands that are not time-overlapped and thus, reducing total amount of bandwidth consumed in a network. An effective and fast heuristic algorithm was developed to route a pair of primary and backup paths for every scheduled LSP demand received. The algorithm proposed is much time efficient than the integer linear programming (ILP) method. By utilizing the characteristics of scheduled LSP demands, the heuristic algorithm proposed performs well and its performance is very close to that of the ILP.

Future works can be done to improve the performance of path protection routing in MPLS networks. The algorithms proposed for multipath protection of dynamic demands use simple shortest path computation in order to provide fast routing in distributed networks. For centralized control networks and demands that can be known in advance, efficient sophisticated algorithms can be developed to provide near optimal performance. The heuristic algorithm proposed for scheduled demands gives good performance in centralized control networks. Developing efficient heuristic algorithms for scheduled LSP demands in distributed networks is a topic for future study.

References

[1] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Technical report, IETF, RFC 3031, January 2001.

[2] http://www.netcraftsmen.net/welcher/papers/mplsintro.html.

[3] http://www.convergedigest.com/bandwidth/archive/010910tutorial-rgallaher1.htm.

[4] Murali Kodialam and T. V. Lakshman. Dynamic Routing of Bandwidth Guaranteed Paths with Restoration. In *Proceedings of IEEE INFOCOM 2000*, March 2000.

[5] Chunming Qiao and Dahai Xu. Distributed Partial Information Management (DPIM) Schemes for Survivable Networks - Part I. In *Proceedings of INFOCOM 2002*, pages 302–311, June 2002.

[6] Dahai Xu, Chunming Qiao, and Yizhi Xiong. An Ultra-fast Shared Path Protection Scheme - Distributed Partial Information Management, Part II. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 344–353, Washington, DC, USA, 2002. IEEE Computer Society.

[7] Yizhi Xiong, Dahai Xu, and Chunming Qiao. Achieving Fast and Bandwidth-Efficient Shared-Path Protection. *Journal of Lightwave Technology*, 21(2), Feb 2003.

[8] Dahai Xu, Yizhi Xiong, and Chunming Qiao. Novel algorithms for Shared Segment Protection. *IEEE Journal on Selected Areas in Communications*, 21(8), October 2003.

[9] Murali Kodialam and T. V. Lakshman. Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels Using Aggregated Link Usage Information. In *Proceedings of IEEE INFOCOM 2001*, pages 376–385, April 2001.

[10] K. Wu, L. Valcarenghi, and A. Fumagalli. Restoration Schemes with Differentiated Reliablity. In *Proceedings of IEEE International Conference on Communications, 2003*, volume 3, pages 1968–1972, Anchorage, Alaska, May 2003.

[11] Ashish Gupta, B.N. Jain, and Satish Tripathi. QoS Aware Path Protection Schemes for MPLS Networks. In *Proceedings of International Conference of Computer Communications*, August 2002.

[12] J. L. Marzo, E .Calle, C. Scoglio, and T. Anjali. Adding QoS Protection in Order to Enhance MPLS QoS Routing. In *Proceedings of IEEE International Conference on Communications, 2003*, volume 3, pages 1973–1977, Anchorage, Alaska, May 2003.

[13] Israel Cidon, Raphael Rom, and Yuval Shavitt. Analysis of Multi-path Routing. *IEEE/ACM Trans. Netw.*, 7(6):885–896, 1999.

[14] A.Elwalid, C.Jin, S.Low, and I.Widjaja. MATE: MPLS Adaptive Traffic Engineering. In *Proceedings of IEEE INFOCOM 2001*, volume 3, pages 1300–1309, 2001.

[15] Youngseok Lee, Yongho Seok, Yanghee Choi, and Changhoon Kim. A Constrained Multipath Traffic Engineering Scheme for MPLS Networks. In *Proceedings of*

IEEE International Conference on Communications, 2002, pages 2431–2436, New York, May 2002.

[16] Yongho Seok, Youngseok Lee, Nakjung Choi, and Yanghee Choi. Fault-tolerant Multipath Traffic Engineering for MPLS Networks. In *IASTED International Conference on Communications, Internet, and Information Technology*, Scottsdale, Arizona, USA, November 2003.

[17] Jeonghwa Song, Saerin Kim, Meejeong Lee, Hyunjeong Lee, and Tatsuya Suda. Adaptive Load Distribution over Multipath in MPLS Networks. In *Proceedings of IEEE International Conference on Communications*, 2003.

[18] Ho Young Cho, Jae Yong Lee, and Byung Chul Kim. Multi-path Constraint-based Routing Algorithms for MPLS Traffic Engineering. In *Proceedings of IEEE International Conference on Communications*, 2003.

[19] Yun-Wen Chen, Ren-Hung Hwang, and Ying-Dar Lin. Multipath QoS Routing with Bandwidth Guarantee. In *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM '01)*, 2001.

[20] Hiroyuki Saito, Yasuhiro Miyao, and Makiko Yoshida. Traffic Engineering Using Multiple Multipoint-to-Point LSPs. In *IEEE INFOCOM*, volume 2, pages 894–901, 2000.

[21] Josue Kuri, Nicolas Puech, Maurice Gagnaire, and Emmanuel Dotaro. Routing and Wavelength Assignment of Scheduled Lightpath Demands. *IEEE Journal on Selected Areas in Communications*, 21(8), October 2003.

[22] Chava Vijaya Saradhi, Lian Kian Wei, and Mohan Gurusamy. Provisioning Fault-Tolerant Scheduled Lightpath Demands in WDM Mesh Networks. In *BROADNETS 2004*, San Jose, California, USA, October 2004.

[23] Changcheng Huang, Vishal Sharma, Ken Owens, and Srinivas Makam. Building Reliable MPLS Networks Using a Path Protection Mechanism. *IEEE Communications Magazine*, 40(3):156–162, March 2002.

[24] Murali Kodialam and T. V. Lakshman. Restorable Dynamic Quality of Service Routing. *IEEE Communications Magazine*, 40(6):72–81, June 2002.

[25] D. O. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan. Extensions to RSVP for LSP Tunnels. *Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-04.txt*, Sept 1999.

[26] R. Guerin, D. Williams, A. Przygienda, S. Kamat, and A. Orda. QoS Routing Mechanisms and OSPF Extensions. In *Proceedings of the IEEE Global Telecommunications Conference, 1997*, 1997.

[27] Koushik Kar, Murali Kodialam, and T. V. Lakshman. Routing Restorable Bandwidth Guaranteed Connections Using Maximum 2-route Flows. In *Proceedings of INFOCOM 2002*, New York, USA, 2002.

[28] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, Inc., 2 edition, 1992.

[29] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[30] Yu Liu. *Spare Capacity Allocation: Model, Analysis and Algorithms*. PhD thesis, University of Pittsburgh, December 2001.

[31] Zhiruo Cao, Zheng Wang, and Ellen W. Zegura. Performance of Hashing-Based Schemes for Internet Load Balancing. In *IEEE INFOCOM 2000*, pages 332–341, Tel-Aviv, Israel, March 2000.

[32] Gero Dittmann and Andreas Herkersdorf. Network Processor Load Balancing for High-Speed Links. In M.S. Obaidat, F. Davoli, I. Onyuksel, and R. Bolla, editors, *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002)*, pages 727–735, San Diego, July 2002. CA Simulation Councils, Inc.

[33] Esmael Dinan, Bijan Jabbari, and Daniel O. Awduche. Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks. In *IEEE International Conference on Communications (ICC-2000)*, volume 3, pages 1604–1608, New Orleans, Louisiana, June 2000.

[34] Youngseok Lee, Yongho Seok, Yanghee Choi, and Changhoon Kim. Dynamic Constrained Multipath Routing for MPLS Networks. In *IEEE ICCCN 2001*.

[35] Ka-Cheong Leung and Victor O. K. Li. Flow Assignment and Packet Scheduling for Multipath Networks. In *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM '99)*, volume 1, pages 246–250, Rio de Janeiro, RJ, Brazil, December 1999.

[36] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Reinhart, and Winston, New York, NY, 1976.

[37] Yu Liu, D. Tipper, and P. Siripongwutikorn. Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing. In *Proceedings of INFOCOM 2001*, pages 699–708, 2001.

[38] C. Li, S. T. McCormick, and D. Simchi-Levi. Finding Disjoint Paths with Different Path Costs: Complexity and Algorithms. *Networks*, 22:653–667, 1992.

[39] Advanced Networking for Research and Education [Online]. Available: http://abilene.internet2.edu/.

[40] Balázs Gábor Józsa and Márton Makai. On The Solution of Reroute Sequence Planning Problem in MPLS Networks. *Computer Networks*, 42(2):199–210, 2003.

[41] Itu-t: Qos routing and related traffic engineering methods - capacity management methods. recommendation e. 360, ca. (2002).

[42] Balázs Gábor Józsa, Dániel Orincsay, and Levente Tamási. Multi-hour Design of Dynamically Reconfigurable MPLS Networks. In *NETWORKING*, pages 502–513, 2004.

[43] Balázs Gábor Józsa, Dániel Orincsay, and András Kern. On the Use of Routing Optimization for Virtual Private Network Design. In *Proceeding of the 7th IFIP Working Conference on Optical Network Design & Modelling (ONDM'2003)*, pages 865–880, Budapest, Hungary, February 2003.

[44] Immanuel M. Bomze., Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. *'The Maximum Clique Problem', Handbook of Combinatorial Optimization (Supplement Volume A).* Kluwer Academic Publishers, Boston, MA, 1999.

[45] R.M. Karp. *'Reducibility among Combinatorial Problems', Complexity of Computer Computations*. Plenum Press, New York, 1972.

[46] David S. Johnson. Approximation Algorithms for Combinatorial Problems. In *STOC '73: Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49. ACM Press, 1973.

[47] Lecky J. E. and R.G. Murphy O. J.and Absher. Graph Theoretic Algorithms for The PLA Folding Problem. *IEEE Trans. Computer-Aided Design*, 8(9):1014C1021, 1989.

[48] Tomita E. and Fujii T. Efficient Algorithms for Finding A Maximum Clique and Their Experimental Evaluation. *Trans. IEICE*, pages 221–228, 1985.

[49] Carraghan R. and Pardalos P. M. An Exact Algorithm for The Maximum Clique Problem. *Oper. Res. Lett.*, 9:375–382, 1990.

[50] T. Soule and J. A. Foster. Using Genetic Algorthms to Find Maximum Cliques (Tech. Rep. No. LAL 95-12). Technical report, Moscow: Department of Computer Science, University of Idaho, 1995.

[51] Elena Marchiori. A Simple Heuristic Based Genetic Algorithm for The Maximum Clique Problem. In *SAC '98: Proceedings of the 1998 ACM symposium on Applied Computing*, pages 366–373, New York, NY, USA, 1998. ACM Press.