University of Alberta

Optimal LSP Capacity and Flow Assignment Using Traffic Engineering in MPLS Networks

By

Weidong Lu ©

A thesis submitted to the faculty of graduate studies and research in partial fulfillment of the requirements of the degree of **Master of Science.**

Department of Electrical and Computer Engineering

Edmonton, Alberta

Fall 2004

Your file Votre référence
ISBN: 0-612-95806-X
Our file Notre référence
ISBN: 0-612-95806-X

Acknowledgements

I would like to thank my parents and my wife. Every time in my life, they are always the source of my power and courage.

During the past few years, Dr. M. Mandal has given me countless advice, guidance, and encouragement. I would like to appreciate his supervisory efforts. I would also like to thank all the members of the Multimedia Computing and Communications Laboratory and other schoolmates in the Department of Electrical and Computer Engineering, and the Department of Computer Science, University of Alberta. When talking to them, I always obtained the good ideas and help.

I would further like to thank all my friends in China, Canada and all around the world. Their friendship gives me so much happiness and I am deeply missing them.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ARIS	Aggregate Route-based IP Switching
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
CA	Capacity Assignment
CFA	Capacity and Flow Assignment
Co-NP	NO answers checkable in polynomial time
Co-NP-complete	the hardest problems in Co-NP
CoS	Class of Service
CR	Constraint-based Routing CR
CR-LDP	Constraint-based Routing Label Distribution Protocol
CSPF	Constraint Shortest Path First
CVP	Conventional Virtual Path
Diffserv	Differentiated Services
DWDM	Dense Wavelength Division Multiplexing
ER	Explicit Route
FA	Flow Assignment
FCFS	First-come First-served
FEC	Forwarding Equivalence Class
GAMS	General Algebraic Modeling System
GCFA	Generic Capacity and Flow Assignment
GFA	Generic Flow Assignment
IETF	Internet Engineering Task Force
IFMP	Ipsilon's Flow Management Protocol
IGP	Interior Gateway Protocol
Intserv	Integrated Services
IPv4	IP version 4
L2-CTS	Layer 2 Cut-Through Switching
LDP	Label Distribution Protocol
LIFO	Last-in First-out
LP	Linear Programming
LSA	Link-State Advertisement
LSP	Label Switched Path
LSR	Label Switch Router
LUR	Link Utilization Ratio
MCF	Multi-Commodity Flow
MF	Multi Field
MFDL	Minimum First Derivative Length
MMPP	Markov Modulated Poisson Process
MPLS	Multiprotocol Label Switching
NLP	Nonlinear Programming
NP	YES answers checkable in polynomial time
NP-complete	the hardest problems in NP

OD	Origin and Destination
OSPF	Open Shortest Path First
P	Solvable in polynomial time
P-complete	the hardest problems in P to solve on parallel computers
PHB	Per-Hop Behavior
PWE3	Pseudo Wire Emulation Edge to Edge
PPVPN	Provider Provisioned Virtual Private Network
QoS	Quality of Service
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol with Traffic Engineering Extension
SDLUR	Standard Deviation of Link Utilization Ratio
SLA	Service Level Agreement
SONET/SDH	Synchronous Optical Network/Synchronous Digital Hierarchy
SP	Stochastic Programming
TCA	Traffic Conditioning Agreement
TE	Traffic Engineering
TLV	Type, Length, Value
TOS	Type of Service
TTL	Time-to-Live
UBCP	Upper Bound of Congestion for a Path
VP	Virtual Path
VPN	Virtual Private Network
WS	Wait and See

Chapter 1

Introduction

The overwhelming growth of the IP networks and the growing popularity of real-time applications set new challenges to the networking community. However, the IP protocol was not designed to provide a guaranteed *Quality of Service* (QoS). As a result, degradation of throughput, long delay, and packet losses occur occasionally in IP networks. In such situations, minimizing the effects of congestion by optimizing the performance of the operational networks becomes more critical. The *Traffic Engineering* (TE) offers service providers a means for performance optimization and bandwidth provisioning. The primary goal of TE in the service provider's networks is to satisfy the customer's traffic demands and minimize congestion while simultaneously optimizing the network resource utilization.

Multiprotocol Label Switching (MPLS), developed by the *Internet Engineering Task Force* (IETF), is an emerging technology that provides QoS as well as TE features in IP networks [1]. It reduces the complexity of packet forwarding in the IP networks, and achieves the simplified connection-oriented forwarding characteristics of Layer 2 switching technologies while retaining the equally desirable flexibility and scalability of Layer 3 routing [2]. When an IP packet enters an MPLS domain, it is assigned a small fixed-length label that specifies the path and its priority. The path along which MPLS packet traverses is called a *Label Switched Path* (LSP). An LSP is defined by the transition in label values, as the label is swapped at each *Label Switch Router* (LSR).

Since the mapping between labels is constant at the LSR, the LSP is actually determined by the initial label value at the ingress LSR. MPLS forwards IP packets to their destination with more efficiency without the need for routers to perform an address lookup for every packet. Hence, the bottleneck of the traditional Layer 3 routing in terms of high speed transmission is solved.

In recent years, several works have been done to efficiently combine Layer 2 switching and Layer 3 routing for fast IP packets transmission. An LSR model presented in [3] focuses on the IP datagram transmission delay in both Layer 2 and Layer 3. Meanwhile, several routing algorithms [4-9] have been proposed for MPLS TE to establish bandwidth guaranteed paths and utilize existing network resources.

In MPLS networks, the objective of traffic control is to achieve a good network efficiency while meeting the users' QoS requirements. Generally, there are two types of traffic controls, which are known as *flow control* and *congestion control*. Flow control ensures that the sender's data rate never exceeds the receiver's data rate. On the other hand, congestion control tries to reduce the possibility of congestion within the network.

From the network level point of view, the LSP distribution in MPLS networks is a critical topic of traffic control. The LSP distribution can be regarded as an abstraction of traffic flows in the network. Accordingly, the LSP distribution problem indeed is a virtual network optimization problem at the flow level. When traffic flows are optimized, the network congestion can be prevented potentially.

In MPLS networks, the LSP distribution problem includes two parts: LSP routing and LSP capacity allocation. By nature, these two parts are interdependent, and an optimal solution can only be obtained by solving LSP routing and LSP capacity allocation jointly. However, in general, the joint optimization problem is very complex. Even in its simplest setting, the LSP distribution problem contains nonlinearities in the objective function, and is very difficult to solve analytically. Therefore, most available solutions are based on a two-phase approach: find capacities with the fixed routing and find the routing with fixed capacities.

1.1 Objective of the Thesis

In this thesis, we treat the TE issue by setting up different types of LSPs through the MPLS network. We first propose an analytical LSP model, in which LSPs are divided into two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing. Using this model, we can achieve Layer 2 simplified connection-oriented forwarding while retaining Layer 3 routing flexibility and scalability.

In order to reduce the overall operations cost, we need to prevent the situation where some parts of the network are over-utilized (congested), while other parts under-utilized. As a result, the optimization of the traffic flow distribution on a given network is an important goal of the TE. In this thesis, we apply the *Generic Flow Assignment* (GFA) model [10] to obtain a new LSP traffic flow distribution objective function in order to minimize the overall weighted delay for the whole network transmission. It is a nonlinear objective function which is based on load dependant parameters. This is feasible since the traffic oscillation in conventional data networks is no longer dominant in MPLS networks [11]. For a quasi-static topology network, the nonlinear model usually can handle the congestion better than the linear model. This nonlinear objective function is expected to distribute the traffic assignment on each link more evenly in the whole network.

Note that the network modeling concept is based on the link-node incidence in the GFA model. In MPLS networks, however, it is more appropriate to use the concept of path-node incidence for the LSP traffic flow distribution. It is natural to use path entities to replace link entities, where the traffic flow through each LSP may belong to a specific traffic class according to multi-class traffic flows in MPLS networks.

1.2 Main Contributions

The main contributions of this thesis are as follows [12]:

- A new analytical L2-CTS LSP model for LSP optimization using TE in MPLS networks. The model combines two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing.
- A nonlinear LSP traffic flow distribution optimization objective function to minimize the overall network transmission weighted delay and optimize the traffic flow distribution across different LSPs.

1.3 Thesis Organization

This thesis is organized as follows: In Chapter 2, the background knowledge about MPLS, QoS, and TE is reviewed. The issues associated with the LSP optimization in MPLS networks, queueing models analysis, and the fundamental theory of computational complexity are reviewed as well. In Chapter 3, the proposed new techniques for LSP optimization using TE in MPLS networks are presented, including the analytical *Layer 2 cut-through switching* (L2-CTS) LSP model and the incorporated nonlinear LSP traffic flow distribution objective function. In Chapter 4, the LSP system resulting from the L2-CTS LSP model and the LSP traffic flow distribution objective function is presented, and compared with the LSP system resulting from the conventional flow assignment model. Two different prototype networks are used for the performance evaluation, and the simulation results are reported. The characterization of numerical results and the computation complexity of the proposed techniques are also analyzed. In Chapter 5, the conclusions and the future research directions are included.

Chapter 2

Background Review

The theory and design of the MPLS network has been advanced significantly in the last 3-4 years. In this Chapter, we present a brief review of different aspects of MPLS networks. The fundamentals of MPLS are reviewed in section 2.1, followed by the review of *Quality of Service* (QoS) and MPLS *Traffic Engineering* (TE) in sections 2.2 and 2.3, respectively. The review of LSP optimization in MPLS networks is then presented in section 2.4. The queueing models and the computational complexity analysis are presented in sections 2.5 and 2.6, respectively.

2.1 Fundamentals of the MPLS Network

In this section, we present a brief review of the fundamental concepts and components of MPLS networks. We explain the working of MPLS networks, and several concepts such as the explicit route, the constraint-based routing, and label distribution protocols.

2.1.1 Challenges to Conventional IP Networks

The exponential growth of both the number of users of the Internet and their bandwidth requirements has placed increasing demands on the service providers' networks. To meet the growing demand for bandwidth, service providers need higher performance switching and routing products. In addition, service providers need to be concerned with the scalability, which can be defined loosely as the ability to grow the network in all dimensions (e.g., increased numbers of nodes, more flows passing through a given node) without finding the insurmountable problem. Furthermore, there is an ever-growing need for new routing functionality, both to handle the growth itself and to meet the evolving needs of the growing user population [1].

It is difficult for the conventional IP network to accomplish these challenges as the IP protocol was not designed to provide a guaranteed QoS. All of these problems lead directly to the key motivations for the development of the label switching approaches.

2.1.2 Overview of MPLS

MPLS is the industry-standard approach developed by the *Internet Engineering Task Force* (IETF) to reduce the complexity of forwarding in IP networks. An overview of MPLS is presented in [13], and its main applications are described in [1]. It is particularly an approach for achieving the simplified connection-oriented forwarding characteristics of Layer 2 switching technologies while retaining the equally desirable flexibility and scalability of Layer 3 routing [2]. By combining the best of link layer switching and network layer routing, MPLS introduces a new forwarding paradigm for IP networks, and brings connection-oriented properties. This is similar to that of the TE capabilities of *Asynchronous Transfer Mode* (ATM) to IP networks, but in a very scalable and cost effective way. The label based switching methods allow routers to make forwarding decisions based on the contents of a simple label, rather than by performing a complex route lookup based on the destination IP address. However, this initial goal of label based switching, in which to bring the speed of Layer 2 switching to Layer 3 routing, is no longer perceived as the main benefit, since Layer 3 switches (ASIC-based routers) are able to perform route lookups at sufficient speeds to support most interface types.

In fact, MPLS brings several other benefits to the IP-based networks which are as follows:

- *Traffic Engineering* (TE): The ability to set up a path through which the traffic will traverse, and the ability to set performance characteristics for different classes of traffic.
- *Virtual Private Networks* (VPNs): Using MPLS, service providers can create IP tunnels throughout their network, without the need for encryption or end-user applications.
- Layer 2 transport: New standards defined by the IETF's PWE3 and PPVPN working groups allow service providers to carry Layer 2 services, including Ethernet, Frame Relay and ATM over an IP/MPLS core.
- Elimination of multiple layers: Most carrier networks typically employ an overlay model where SONET/SDH is deployed at Layer 1, ATM is used at Layer 2 and IP is used at Layer 3. Using MPLS, the carriers can migrate many of the functions of the SONET/SDH and ATM control plane to Layer 3, thereby simplifying network management and network complexity. Eventually, the carrier networks may be able to migrate away from SONET/SDH and ATM all-together, which means elimination of ATM's inherent "cell-tax" in carrying IP traffic.

In MPLS networks, a *label* is a short, fixed length, locally significant identifier which is used to identify a *Forwarding Equivalence Class* (FEC). The label, which is put on a particular packet, represents the FEC to which that packet is assigned. A label may be considered as a shorthand for the packet header, which is used by the router to make an appropriate forwarding decision. In this context, the label is nothing more than a shorthand for an aggregate stream of user data.

The format of the MPLS label is shown in Fig. 2-1. The MPLS label contains the following fields:

- The label field (20-bits): carries the actual value of the MPLS label.
- The *Class of Service* (CoS) field (3-bits): can affect the queueing and discard algorithms applied to the packet as it is transmitted through the network.
- The *Stack* (S) field (1-bit): supports a hierarchical label stack.
- The *time-to-live* (TTL) field (8-bits): provides conventional IP TTL functionality. This is also called a "Shim" header.

20 bits Label	3 bits CoS	1 bit Stack	8 bits TTL

Figure 2-1. MPLS label format.

The 32-bits MPLS label is located after the link layer header and before the network layer header. The path along which the MPLS packets traverse is called a *Label Switched Path* (LSP). An LSP is provisioned using *Label Distribution Protocols* (LDPs). These protocols establish a path through the MPLS network and reserve necessary resources to meet pre-defined service requirements for the path.

An LSP can be contrasted with a traffic trunk. A traffic trunk is an aggregation of traffic flows of the same class which are placed inside an LSP. It is important to emphasize that there is a fundamental distinction between a traffic trunk and an LSP. The LSP through which a trunk traverses can be changed. In this respect, traffic trunks are similar to virtual circuits in ATM and Frame Relay networks. In practice, however, the terms LSP and traffic trunk are often used synonymously.

At each hop across an LSP tunnel through an MPLS domain, the packet gets a new label value that determines an outbound interface to the next hop, and its treatments. The LSP is defined by the transition in label values, as the label is swapped at each *Label Switch Router* (LSR). The LDP establishes an LSP by using a set of procedures to distribute the labels among the LSR peers. Since the mapping between labels is constant at each LSR, the LSP is actually determined by the initial label value at the ingress LSR.

A basic operation of an MPLS network is shown in Fig. 2-2. In this network, Host A is sending IP packets to destination Host C, while Host B is sending IP packets to destination Host D. As IP packets enter the MPLS domain from both sources, the ingress LSR 1 typically determines which LSP to use for each packet, and attaches a label to the outgoing packet accordingly. Note that it is the label value at the ingress router that actually determines which LSP to travel for the packet. LSR 1 then forwards the packet via the appropriate interface for the selected LSP. When the intermediate LSR 2 receives a packet, it decides, based on the incoming interface and the label value, the outgoing interface and the label value with which to forward the packet to the next hop. Thus, according to Fig. 2-2, at the ingress router, IP packets from origin Host A to destination Host C are mapped into LSP 1 by attaching label 13. The intermediate LSR 2 simply forwards each packet with label 13 to LSR 3 after swapping with the new label 65. At the same time, packets from origin Host B to destination Host D are mapped into LSP 2 by attaching label 22. As the packets traverse LSP 2, the intermediate LSR 2 examines the label in the MPLS header of each packet, and looks up its MPLS forwarding table to match for the incoming label 22. LSR 2 then swaps label 22 with an outgoing label 16 before it forwards to the egress LSR 3. Finally, at the egress LSR 3, all labels are removed, and packets are then forwarded using just the traditional IP forwarding paradigm to their respective destinations.

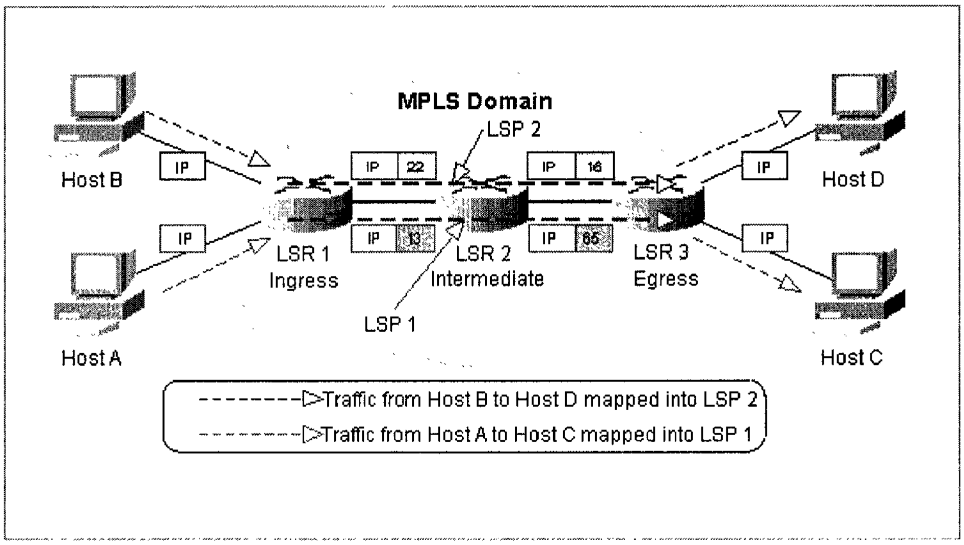


Figure 2-2. A simplified schematic example of an MPLS network.

Since MPLS allows the hierarchy of routing knowledge known as *label stack*, it is possible to have different LSPs at different levels of labels for a packet to reach its destination. An example of the hierarchy of routing knowledge in an MPLS network is shown in Fig. 2-3.

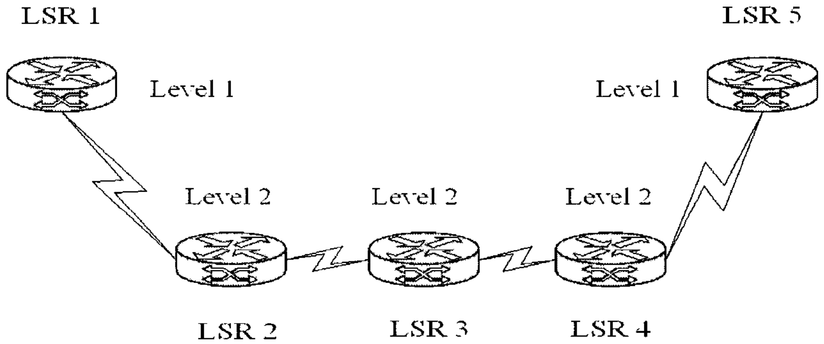


Figure 2-3. The hierarchy of routing knowledge in an MPLS network.

In Fig. 2-3, LSR 1 to LSR5 are the LSRs that a packet p must travel to reach its destination. The numbers "1" and "2" are label stack depth. LSR 1 and LSR 5 are border gateway routers and LSR 2, LSR 3 and LSR 4 are the interior gateway routers. For the purpose of label forwarding, LSR 1 and LSR 5 are peers at the border gateway level and LSR 2, LSR 3, LSR 4 are peers at the interior gateway level. When LSR 1 receives the packet p with a label that is "1" level deep, it will swap packet p's label with a corresponding label that will be used by LSR 5. In addition, since the packet has to travel through LSR 2, LSR 3 and LSR 4, LSR 1 will push a new label onto the label stack carried by the packet p, and the label stack level is now "2". Hence, we have two LSPs here: one is at level 1 from LSR 1 to LSR 5 and the other is at level 2 from LSR 2 to LSR 4. The use of a hierarchy of routing knowledge allows complete isolation of the interior routers within a routing domain from the inter-domain routing. Thus, it improves the stability, convergence and scalability of routing.

In order to explain further how the labels are established and switched, and how an LSP is set up in an MPLS network, a detailed example is shown in Fig. 2-4. Since the label switching control component uses both local and remote bindings to populate its forwarding table with incoming and outgoing labels, the *downstream label binding* method is chosen here, which means labels from the local binding are used as the incoming labels and labels from the remote binding are used as the outgoing labels. Note that by using the downstream label binding, the flow of the binding information of the label is opposite to the flow of packets in the direction. The detailed steps are as follows:

Step 1: The bottom non-MPLS (customer) router has Class C networks 192.1.1.0 /24, 192.1.2.0 /24 out the Ethernet 0 interface. The routing table in this step tracks the routing prefix, the outgoing interface, the next hop router, and other information. The light blue arrow suggests that an ordinary routing update advertises the routes to the edge LSR above.

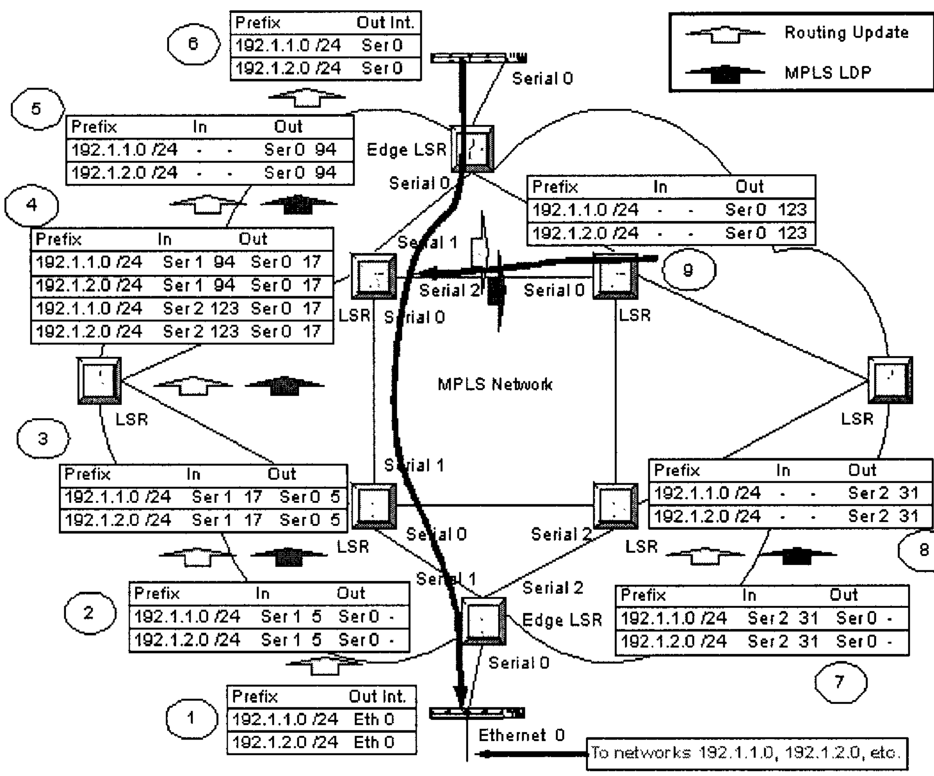


Figure 2-4. A detailed example of an MPLS network.

Step 2: The routes are advertised to the LSR above and to the left of the edge LSR. Using LDP, the router selects a free (unused) label, 5, and advertises it to the upstream neighbor. Note that the hyphens in the Out column of the routing table mean that all labels are to be popped (removed) in forwarding to the non-LSR below. Thus, a frame received on Serial 1 with label 5 is to be forwarded out Serial 0 with no label. The red arrow suggests LDP communicating the use of label 5 to the upstream LSR.

Step 3: The LSR has learned routes to the two prefixes we are tracking. It advertises the routes upstream. When LDP information is received, it records the use of label 5 on the outgoing interface Serial 0 for the two prefixes we are tracking. It then allocates label 17 on Serial 1 for this FEC, and uses LDP to communicate this to the upstream LSR. Thus, when label 17 is received on Serial 1, it is replaced with label 5 and the frame sent out Serial 0.

Steps 4 and 5: Proceed similarly. Note that there will be no labels received at the top edge LSR, since the top router is not an MPLS participant, as we can see from its routing table (no labels) in Step 6. The dark blue arrow shows that the LSP has now been established. The table in Step 4 is bigger since this LSR has sent the routing and LDP information to the LSR to its right.

Step 7: A routing advertisement might also be sent out interface Serial 2 from the edge LSR at the bottom. It can also use LDP to tell the upstream LSR to use label 31 to deliver packets rapidly to the destinations we are tracking here.

Step 8: When LDP information is received in this LSR, it records the use of label 31 on the outgoing interface Serial 2. The use of LDP to communicate to the upstream LSR is not specified here (as it is similar to the processing in Step 3).

Step 9: Here we have bindings that have passed from the left LSR to the right one. The right one uses label 123 for our two prefixes. Note that multiple flows can end up merging: frames bearing label 94 on Serial 1 or label 123 on Serial 2 all get re-labeled with label 17 and sent out Serial 0. This indicates the multipoint-to-point behavior of MPLS.

2.1.3 Explicit Route and Constraint-Based Routing

An *Explicit Route* (ER) is generally understood as a precise sequence of steps from the ingress to the egress nodes. An LSP in the MPLS can be set up to follow an explicit path (i.e., a list of IP addresses). However, it does not need to specify the route fully explicitly. For example, the route could specify only the first few hops. After the last explicitly specified hop has been reached, the routing of the LSP will proceed using hop-by-hop routing. A component of an ER may also be less precisely specified. A collection of nodes, known as "Abstract Nodes", may be presented as a single step in the route, for instance, by using an IP prefix rather than a precise address. The LSP must be routed to some nodes within this abstract node as the next hop. The route may contain several hops within the abstract node before emerging to the next hop specified in the ER.

An ER may also contain the identifier of an *Autonomous System* (AS). This allows the LSP to be routed through an area of the network that is out of the administrative control of the initiator of the LSP. The route may contain several hops within the AS before emerging to the next hop specified in the ER.

An ER may be classified as "strict" or "loose". A strict route must contain only those nodes, abstract nodes or ASs specified in the ER, and must use them in the order specified. A loose route must include all of the hops specified, and must maintain the order, but it may also include additional hops as necessary to reach the hops specified. Once a loose route has been established, it can be modified, or can be "pinned" so that it does not change.

Explicit routing is particularly useful to force an LSP down a path that differs from the one offered by the routing protocol. It can be used to distribute traffic in a busy network, to route around network failures or hot spots, or to provide pre-allocated back-up LSPs to protect against network failures.

The route that an LSP may take can be constrained by many requirements selected at the ingress LSR. An ER is an example of a constraint-based route where the constraint is the order in which intermediate LSRs may be reached. Other constraints can be imposed by a description of the traffic flow, and may include bandwidth, delay, resource class and priority.

There are two main approaches to specify routes based on the constraints [14]: one approach is for the ingress LSR to calculate the entire route based on the constraints and information that it has about the current state of the network. This leads it to produce an ER that satisfies the constraints. The other approach is a variation of hop-by-hop routing where, at each LSR, the next hop is calculated using information held at that LSR about local resource availability.

These two approaches are combined if the information about parts of the route is unavailable (e.g., it traverses through an AS). In this case, the route may be loosely specified in part, and explicitly routed using the constraints where necessary.

Constraint-based routing computes routes that are subject to constraints such as bandwidth and administrative policy. Since constraint-based routing considers more than network topology in computing routes, it may choose a longer but lightly loaded path over the heavily loaded shortest path. Hence, the traffic flows may be distributed more evenly across the network [15].

An example of constraint-based routing is shown in Fig. 2-5. Here, the shortest path between LSR A and LSR C is through link A-C with *Interior Gateway Protocol* (IGP) metric m=1. But because the reservable bandwidth on the shortest path is only (622-600)=22 Mbps (see Fig. 2-5), when constraint-based routing tries to find a path for an LSP of 40 Mbps, it will select path A-B-C instead, since the shortest path does not meet the bandwidth constraint.

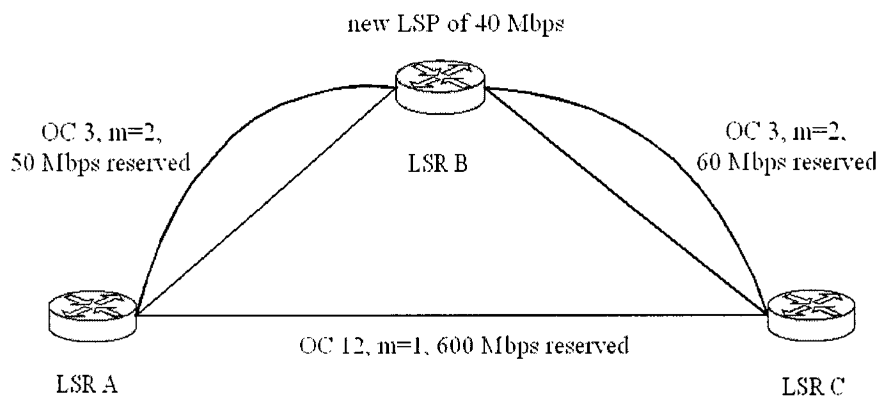


Figure 2-5. Constraint-based routing.

Constraint-based routing can be online or offline. With online constraint-based routing, LSRs may compute paths for LSPs at any time. With offline constraint-based routing, an offline server computes paths for LSPs periodically (on hourly/daily basis). LSPs are then configured to take the computed paths.

2.1.4 Label Distribution Protocol

Label Distribution Protocol (LDP) is a major part of the MPLS. Similar mechanisms for label exchange exist in vendor implementations, such as *Ipsilon's Flow Management Protocol* (IFMP), IBM's *Aggregate Route-based IP Switching* (ARIS), and Cisco's *Tag Distribution Protocol*. The LDP and labels are the foundation of the label switching.

Some basic characteristics of LDP are as follows:

- It provides an LSR "discovery" mechanism to enable LSR peers to find each other and establish the communication.

- It defines four classes of messages: DISCOVERY, ADJACENCY, LABEL ADVERTISEMENT, and NOTIFICATION messages.
- It runs over TCP to provide reliable delivery of messages (with the exception of DISCOVERY messages).

By using LDP, label distribution and assignment may be performed in several different modes:

- Unsolicited downstream versus downstream-on-demand label assignment.
- Order versus independent LSP control.
- Liberal versus conservative label retention.

To ensure that LSPs can be used, the forwarding tables at each LSR must be populated with the mappings from {incoming interface, label value} to {outgoing interface, label value}. This process is called *LSP setup*, or *label distribution*. Note that the MPLS architecture document does not mandate a single protocol for the distribution of labels between LSRs. In fact, it specifically allows using multiple protocols in different scenarios.

Several different approaches to label distribution are generally used depending on the requirements of the hardware that forms the MPLS network, and the administrative policies used on the network. The underlying principles are that an LSP is set up either in response to a request from the ingress LSR (*downstream-on-demand*), or pre-emptively by LSRs in the network, including the egress LSR (*downstream unsolicited*). It is possible for both to take place at once and for the LSP to meet in the middle.

In all cases, labels are allocated from the downstream direction (where downstream refers to the direction of data flow). Thus, in the example as shown in Fig. 2-6, LSR D informs LSR B that LSR B should use label 47 on all packets for host Z. LSR B allocates a new label (21), enters the mapping in its forwarding table, and informs LSR A that it should use label 21 on all packets for host Z.

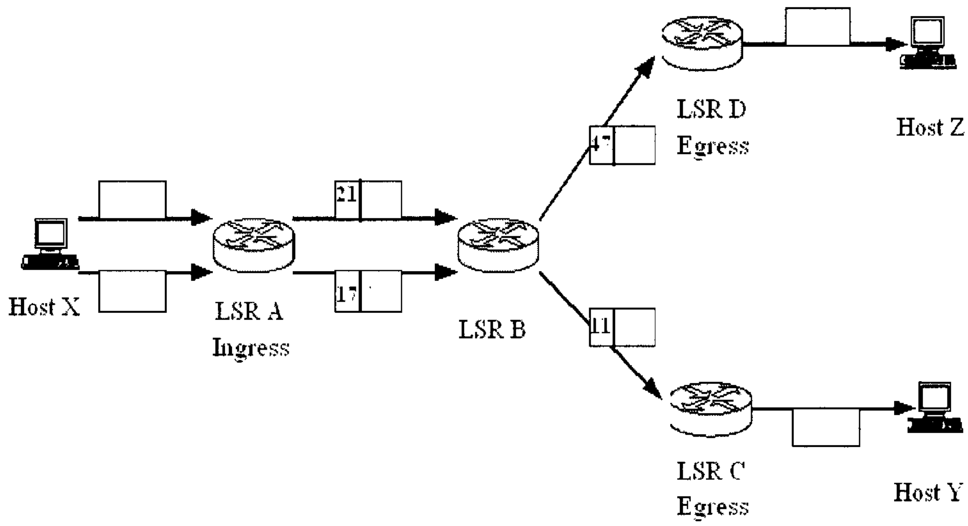


Figure 2-6. Label distribution protocol.

Some possible options for controlling the setting up of LSPs, and the protocols that can be used to achieve them, are described in the following:

- Hop-by-hop label assignment is the process by which the LSP setup requests are routed according to the next-hop routing towards the destination of the data. The LSP setup could be initiated by updates to the routing table, or in response to a new traffic flow. The IETF MPLS working group has specified (but not mandated) the LDP as a protocol for hop-by-hop label assignment. *Constraint-based Routing Label Distribution Protocol* (CR-LDP) and *Resource Reservation Protocol* (RSVP) are two main signaling protocols that can be used.
- In *downstream unsolicited* label distribution, the egress LSR distributes the label to be used to reach a particular host. The trigger for this will usually be

the new routing information received at the egress node. Additionally, if the label distribution method is *ordered control*, each upstream LSR distributes a label further upstream. This effectively builds a tree of LSPs rooted at each egress LSR. LDP is currently the only protocol suitable for this mode of label distribution.

- Once LSPs have been established across the network, they can be used to support new routes as they become available. As the routing protocols (e.g., BGP) distribute the new routing information upstream, they can also indicate which label (i.e., which LSP) should be used to reach the destinations to which the route refers.
- If an ingress LSR wants to set up an LSP that does not follow the next-hop routing path, it must use the LDP that allows specification of an ER. This requires *downstream-on-demand* label distribution. CR-LDP and RSVP are two protocols that provide this function.
- An ingress LSR may also want to set up an LSP that provides a particular level of service by, e.g., reserving resources at each intermediate LSR along the path. In this case, the route of the LSP may be constrained by the availability of resources and the ability of nodes to fulfill the QoS requirements. CR-LDP and RSVP are two protocols that allow *downstream-on-demand* label distribution to include requests for specific service guarantees.

As mentioned above, *Constraint-based Routing Label Distribution Protocol* (CR-LDP) is a set of extensions to LDP specifically designed to facilitate constraint-based routing of LSPs. It uses TCP sessions between LSR peers and sends label distribution messages along the sessions. This allows it to assume reliable distribution of control messages.

Resource Reservation Protocol with Traffic Engineering Extension (RSVP-TE) uses a message exchange to reserve resources across a network for IP flows. The extensions to RSVP for LSP tunnels enhance the generic RSVP so that it can be used to distribute MPLS labels. It uses IP datagrams (or UDP at the margins of the network) to communicate between LSR peers. It does not require the maintenance of TCP sessions, but as a consequence of this it must handle the loss of control messages.

Table 2-1. Comparison of CR-LDP and RSVP.

	CR-LDP Support	**RSVP Support**
Transport	TCP	Raw IP
Security	Yes	Yes
Multipoint-to-Point	Yes	Yes
Multicast Support	No	No
LSP Merging	Yes	Yes
LSP State	Hard	Soft
LSP Refresh	Not needed	Periodic, hop-by-hop
High Availability	No	Yes
Re-routing	Yes	Yes
Explicit Routing	Strict and loose	Strict and loose
Route Pining	Yes	Yes, recording path
LSP Pre-emption	Yes, priority based	Yes, priority based
LSP Protection	Yes	Yes
Shared Reservations	No	Yes
Traffic Parm Exchange	Yes	Yes
Traffic Control	Forward path	Reverse path
Policy Control	Implicit	Explicit
Layer 3 Protocol Indicated	No	Yes
Resource Class Constraint	Yes	No

CR-LDP and RSVP-TE are both good technical solutions for setting up and managing traffic engineered LSPs. The key differences between CR-LDP and RSVP are the reliability of the underlying transport protocol and whether the resource reservations are done in the forward or reverse direction. Table 2-1 summarizes the main similarities and differences between CR-LDP and RSVP for LSP tunnels.

2.2 Quality of Service

Today's IP networks only provide the best effort service. With the rapid transformation of the Internet into a commercial and entertaining infrastructure, demands for QoS have rapidly increased.

Several types of services are required for various applications. One type provides the predictable service for companies that do business on the Web. Such companies are willing to pay a certain price to make their services reliable and to give their users a better QoS. This type of services may contain a single service class, or it may contain multiple classes such as *gold* and *silver* with decreasing quality. Another service type provides low delay and low jitter services to real-time applications. Finally, the best effort service will remain for those customers who only need connectivity [16].

The necessity of providing the QoS in IP networks is still a debated issue. One group of researchers feel that the fiber and dense wavelength division multiplexing (DWDM) technologies will make bandwidth so abundant and cheap that high QoS will be automatically delivered. Another group feels that no matter how much bandwidth a network can provide, new applications will be created to consume it. Therefore, mechanisms will still be needed to provide QoS. In our opinion, even if the network bandwidth will eventually become abundant and cheap, it is not the reality now. Hence, some simple mechanisms are now definitely needed to provide QoS in IP networks. All major router/switch vendors support this view by providing some QoS mechanisms in their products [17-21].

The IETF has proposed several service models and protocols for providing QoS in IP networks. Notably among them are the Intserv/RSVP and Diffserv. They are briefly reviewed in the subsequent sections.

2.2.1 Integrated Services

The *integrated services* (Intserv) model [22] proposes two service classes in addition to the best effort service. They are: 1) *guaranteed service* [23] for applications requiring the fixed delay bound; and 2) *controlled load service* [24] for applications requiring the reliable and enhanced best effort service. The philosophy of this model is that "there is an inescapable requirement for routers to be able to reserve resources in order to provide special QoS for specific user packet streams, or flows. This in turn requires flow-specific state in the routers" [22].

As mentioned in section 2.1.4, RSVP was invented as a signaling protocol for applications to reserve resources [25]. The signaling process is illustrated in Fig. 2-7. The sender sends a PATH message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. Upon receiving a PATH message, the receiver responds with a RESV message to request resources for the flow. Every intermediate router along the path can reject or accept the request of the RESV message. If the request is rejected, the router will send an error message to the receiver, and the signaling process will terminate. If the request is accepted, the link bandwidth and the buffer space are allocated for the flow and the related flow state information will be installed in the router.

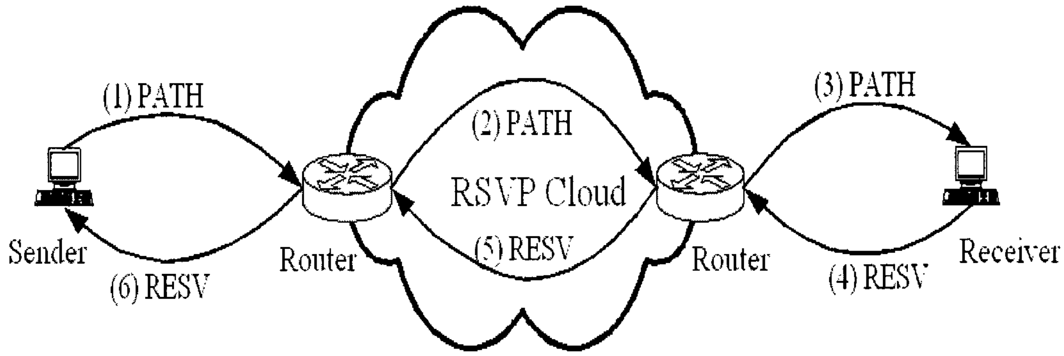


Figure 2-7. RSVP signaling.

Intserv is implemented by four components: signaling protocol (e.g., RSVP), admission control routine, classifier and packet scheduler. Applications requiring guaranteed service or controlled-load service must set up paths and reserve resources before transmitting their data. The admission control routines will decide whether a request for resources can be granted. When a router receives a packet, the classifier will perform a *multi-field* (MF) classification and put the packet in a specific queue based on the classification result. The packet scheduler will then schedule the packet accordingly to meet its QoS requirements.

The Intserv/RSVP architecture is influenced by the work of Ferrari *et al.* [26]. It represents a fundamental change to the traditional Internet architecture, which is established with the assumption that all flow-related state information are available in the end systems [27].

The Intserv architecture has several limitations. First, the amount of state information increases proportionally with the number of flows. This places a huge storage and processing overhead on the backbone routers. Therefore, this architecture does not scale well in the IP core. Secondly, the requirement on routers is high. All routers must support RSVP, admission control, MF classification and packet scheduling. Thirdly, ubiquitous deployment is required for the guaranteed service. Incremental deployment of the controlled load service is possible by deploying controlled-load service and RSVP functionality at the bottleneck nodes of a domain and tunneling the RSVP messages over other part of the domain.

2.2.2 Differentiated Services

Because of the difficulty in implementing and deploying Intserv and RSVP, *differentiated services* (Diffserv) [28] has been introduced. The essence of Diffserv is to divide the traffic into multiple classes, and treat them differently, especially when there is a shortage of resources.

IP version 4 (IPv4) header contains a *type of service* (TOS) byte. Its meaning is previously defined in [29]. Applications can set the left three bits in the TOS byte to indicate the need for low delay or high throughput or low loss rate service. However, the choices are limited. Diffserv renames the TOS octet as *differentiated services field* (DS field) [30] and uses it to indicate the forwarding treatment that a packet should receive. Diffserv also standardizes a number of *per-hop behavior* (PHB) groups [31,32]. Using different classification, policing, shaping and scheduling rules, several classes of services can be provided.

To ensure that a customer can receive differentiated services from its service provider, it must have a *service level agreement* (SLA). An SLA may explicitly or implicitly specify a *traffic conditioning agreement* (TCA) which defines classification rules as well as metering, policing, marking, and shaping rules [28]. An SLA can be static or dynamic. Static SLAs are negotiated on a regular basis, e.g., monthly and yearly.

Customers with dynamic SLAs use a signaling protocol such as RSVP to request for services on demand.

Customers can mark the DS fields of their packets to indicate the desired service or have them marked by the edge routers based on the MF classification. At the ingress router of IP networks, packets are classified, policed and possibly shaped. The classification, policing and shaping rules used at the ingress routers are specified in the TCA. The amount of buffering space required is usually not considered explicitly. When a packet enters one domain from another domain, its DS field may be re-marked, as determined by the SLA between the two domains.

Diffserv only defines DS fields and PHBs. It is the responsibility of service providers to decide what kind of services to provide. Using the classification, policing, shaping and scheduling mechanisms, many services can be provided, such as:

- *Premium* service with reliable, low delay and low jitter delivery.
- *Gold* and *silver* services with reliable and timely delivery.

Note that Diffserv is significantly different from Intserv. First, there are only a limited number of service classes indicated by the DS field. Since resources are allocated in the granularity of class, the amount of state information is proportional to the number of classes rather than the number of flows. Diffserv is therefore more scalable. Secondly, sophisticated classification, marking, policing and shaping operations are only needed at the edge of the networks. The IP core routers only need to implement *behavior aggregate* (BA) classification, and hence, it is easier to implement Diffserv.

In the Diffserv model, incremental deployment is possible for gold and silver services. Diffserv-incapable routers simply ignore the DS fields of the packets and give all packets the best effort service. Even in this case, because gold and silver packets are less likely to be dropped by Diffserv-capable routers, the performance of gold and silver traffic will still be better than the best effort traffic.

2.3 MPLS Traffic Engineering

In this section, the overview of traffic engineering is first presented. The traffic engineering issues in MPLS networks are then reviewed.

2.3.1 Overview of Traffic Engineering

Traffic Engineering (TE) refers to the process of selecting the paths chosen by the traffic in order to balance the traffic load on the various links, routers, and switches in the network. The TE issue is important in the network where multiple parallel or alternate paths are available. A major goal of TE is to facilitate efficient and reliable network operations while simultaneously optimizing the network resource utilization and the traffic performance [33]. The TE is needed in IP networks mainly because current IGPs always use the shortest paths to forward the traffic. It can conserve network resources using shortest paths, but it may also cause the following problems:

- ❖ The shortest paths from different sources overlap at some links, causing congestion on those links.
- ❖ The traffic from a source to a destination exceeds the capacity of the shortest path, while a longer path between these two routers is under-utilized.

By performing the TE in IP networks, the resource utilization and the network performance can be optimized significantly. Revenue can be increased without large investments in upgrading the network infrastructure.

The objective of TE is to compute a path from one given node to another (source routing), such that the path does not violate the constraints (e.g., bandwidth or administrative requirements), and it is optimal with respect to some scalar metric. Once the path is computed, TE (constraint-based routing) is responsible for establishing and maintaining the forwarding state along such a path.

In order to support TE, besides the explicit routing (source routing), the following components should be available:

- ❖ Compute a path at the source by taking into account all the constraints. Therefore, the source need to have all the information either available locally or obtained from other routers in the network (e.g., network topology).
- ❖ Distribute the information about the network topology and attributes associated with links throughout the network. Once the path is computed, there should be a way to support the forwarding along such a path.
- ❖ Reserve network resources and modify link attributes (as the result of certain traffic taking certain routes).

2.3.2 TE Issues in MPLS Networks

MPLS addresses the TE issue by setting up explicit paths through the network using constraint-based routing. The requirements for TE over MPLS are presented in [33].

MPLS TE uses several fundamental technologies, some are listed as follows:

- ❖ *Constraint Shortest Path First* (CSPF) algorithm is used in the path calculation. This is a modified version of the well known SPF algorithm extended to support the constraints.

- ❖ RSVP extension or CR-LDP is used to establish the forwarding state along the path as well as to reserve resources along the path.
- ❖ Enhanced link state IGPs (e.g., OSPF with *Opaque Link-State Advertisements* (Opaque LSAs), IS-IS with *Link-State Packets* (LSPs) TLV (type, length, value)) are used to propagate link attributes in addition to the normal link state information.

MPLS TE dynamically establishes and maintains an LSP tunnel across the MPLS domain using signaling protocols. The two signaling mechanisms (i.e., CR-LDP and RSVP-TE) used for distributing labels across an MPLS domain have been reviewed in section 2.1.4. The LDPs (i.e., CR-LDP and RSVP) determine the path across which the LSP tunnel is established based on its resource requirements and available network resources (e.g., bandwidth). Then the LDPs move the label binding information along a predefined route. At the ingress LSR, the label is assigned to packets as they enter the network. The label binding to packets is done based on the FEC membership. This feature of MPLS allows the scalable aggregation of traffic flows into a single FEC based on their requirements. Besides, it makes the task much simpler for MPLS TE to route the traffic flows across an LSP tunnel by associating the resources required by a given FEC (or LSP) with the actual network capacity and topology.

When a link goes down, it is necessary to reroute all trunks that were routed over this link. Beause the path taken by a trunk is determined by the ingress LSR (i.e., head end LSR) of the MPLS domain, rerouting has to be performed by the head end LSR. To perform rerouting, the head end LSR could rely either on the information provided by IGP or by RSVP/CR-LDP.

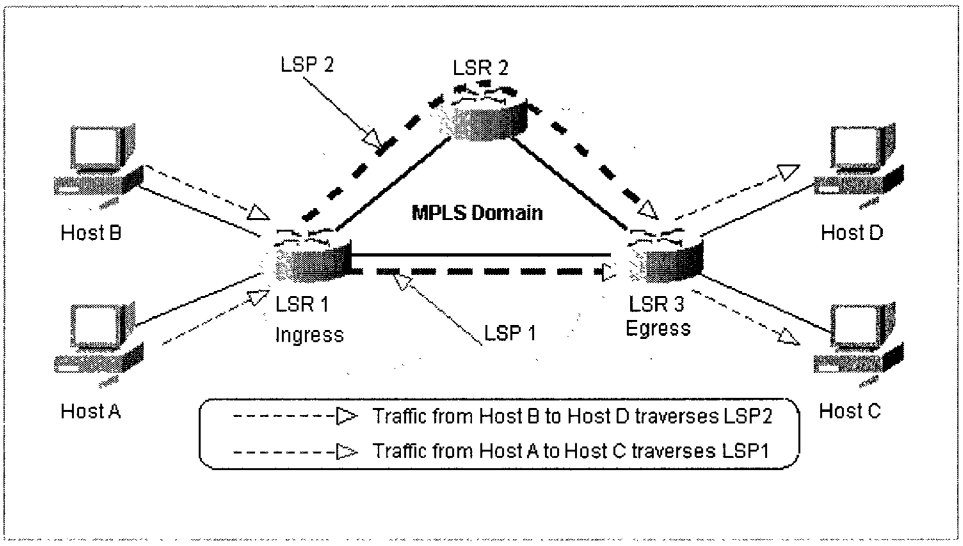


Figure 2-8. Traffic engineering using explicit LSPs.

A simple example shown in Fig. 2-8 explains how explicit LSPs are used to solve the traffic engineering problem. In Fig. 2-8, LSR 1 through LSR 3 are in the MPLS domain, while Host A through Host D are traffic sources and destinations. Assume a 100 Mbps PVC connections among all LSRs. Further assume that the traffic from Host A to Host C is 100 Mbps and that of the traffic from Host B to Host D is 100 Mbps. With MPLS explicit routing capability, the traffic from Host B to Host D can be assigned to LSP 2, which in turn traverses the path LSR 1-LSR 2-LSR 3, while the traffic from Host A to Host C can be made to travel across LSP 1, which consists of path LSR 1-LSR 3. As a result, the traffic flows between hosts one side (in this case, Host A and Host B) and hosts on the other side (Host C and Host D) can be distributed over the network according to their demands, such as bandwidth guarantees, by just establishing different LSPs. This enables achieving the efficient bandwidth utilization as well as a significant performance gain.

2.4 LSP Optimization in MPLS Networks

In this section, a few key concepts such as LSP attributes, MPLS traffic control and LSP optimization are reviewed. Because of the similarities between the LSP distribution in MPLS networks and the *virtual path* (VP) distribution in ATM networks, a few VP optimization schemes are also reviewed.

2.4.1 LSP Attributes

In order to control LSPs effectively, each LSP can be assigned one or more attributes. These attributes will be considered in computing the path for the LSP. Such attributes are summarized in Table 2-2.

Table 2-2. LSP attributes [34].

Attribute Name	Meaning of the attribute
Bandwidth	The minimum requirement on the reservable bandwidth of a path for the LSP to be set up along that path
Path attribute	An attribute that decides whether the path of the LSP should be manually specified or dynamically computed by the constraint-based routing
Setup Priority	The attribute that decides which LSP will get the resource when multiple LSPs compete for it
Holding Priority	The attribute that decides whether an established LSP should be preempted the resource it holds by a new LSP
Affinity (color)	An administratively specified property of an LSP
Adaptability	Whether to switch the LSP to a more optimal path when one becomes available
Resilience	The attribute that decides whether to reroute the LSP when the current path is affected by failure

2.4.2 MPLS Traffic Control

In MPLS networks, the goal of traffic control is to achieve the good network efficiency while meeting the users' QoS requirements. This goal implies two conflicting objectives: high network efficiency and user's QoS guarantee. The heterogeneous traffic in MPLS makes these goals more difficult to achieve.

Generally, there are two types of traffic controls, which are known as *flow control* and *congestion control*. Flow control ensures that the sender's data rate never exceeds the receiver's data rate. On the other hand, congestion control tries to reduce the possibility of congestion within the network. Congestion control can be classified into two categories: *preventive control* and *reactive control* [35]. Reactive control is based on a feedback mechanism, while preventive control is based on the network provisioning.

From the network level point of view, the LSP distribution in MPLS networks is a critical topic of traffic control. The LSP distribution can be regarded as an abstraction of traffic flows in the network. Accordingly, the LSP distribution problem indeed is a virtual network optimization problem at the flow level. When traffic flows are optimized, the network congestion can be prevented potentially.

In MPLS networks, the LSP distribution problem includes two parts: LSP routing and LSP capacity allocation. By nature, these two parts are interdependent, and an optimal solution can only be obtained by solving LSP routing and LSP capacity allocation jointly. However, in general, the joint optimization problem is very complex. Even in its simplest setting, the LSP distribution problem contains nonlinearities in the objective function, and is very difficult to solve analytically [36]. Therefore, most available solutions are based on a two-phase approach: find capacities with the fixed routing and find the routing with fixed capacities. The two-phase approach has two basic forms: iterative or non-iterative. The iterative form solves the two-phase models recursively until the error between the updated solution and the last solution is less than a pre-specified tolerance. The recursive process will converge because there are only a finite number of optimal path flows. However, the non-iterative form tries to establish a set of equations in terms of both capacities and flows. The order of these equations is usually not high, typically being a quadratic system. Therefore, many well-developed algorithms can be adopted to solve it. Note that the non-iterative form differs from the joint approach mentioned above as the joint approach is inherently a one-phase approach.

2.4.3 Review of VP Optimization Schemes

Because of the similarities between the LSP distribution in MPLS networks and the VP distribution in ATM networks, a few selected VP optimization schemes are reviewed in this section.

The VP distribution problem has been studied by numerous researchers. These researchers have tried to solve the VP distribution problem based on different objectives. Anerousis *et al.* [37] proposed a scheme to allocate bandwidth to VPCs based on a given fixed VP topology. This scheme can satisfy nodal constraints on the call processing load and blocking constraints for each *origin and destination* (OD) pair, and support multiple traffic classes. In their scheme, an optimal solution is obtained in terms of balancing the network call throughput and the processing load on the signaling system. However, the scheme is complex and computationally expensive.

Chlamtac *et al.* [38] presented an algorithm to find a system of VP routes for a given set of VP terminators and VP capacity demands. Considering the link capacity is limited in ATM networks, they proposed a scheme to find an optimal VP topology in terms of minimizing the maximum link load in the network. The main assumption here is that if the traffic is distributed in a way that reduces the maximum link load, the possibility of congestion can be decreased and the network robustness can be increased. Based on a given set of VP terminators and VP capacity demands, the algorithm can find an optimal VP topology while minimizing the load and reducing congestion on individual links.

Ahn *et al.* [39] proposed a VP layout design method to solve the VP routing problem based on a flow optimization algorithm with heuristics. The algorithm can find an optimal VP topology in terms of carrying maximum percentage of the offered traffic while minimizing call blocking probability, call setup, switching, and transmission costs. Guidelines for the design of robust VP layouts and the efficient establishment of VCs are also presented. In this VP layout design scheme, only one traffic class is considered.

Ryu *et al.* [40] proposed a heuristic design algorithm for the VP distribution problem based on an equivalent bandwidth concept. An equivalent bandwidth of a set of VCs multiplexed on a VP is defined as an amount of the bandwidth required to achieve a desired QoS. In their scheme, fluid flow approximation and stationary approximation methods [41] are used to calculate the equivalent bandwidth. They have tried to solve this joint VP distribution problem in terms of minimizing network costs. In this scheme, only one class of traffic is considered, and all VCs are considered to be grouped into VPs.

Zheng *et al.* [4] proposed a generic capacity and flow assignment (GCFA) model based on path-node incidence for satellite ATM networks. The proposed GCFA model can be used to solve the VP distribution problem by assuming that all the traffic is carried on VPs. In this model, the link capacity is assumed to be changeable, and *Stochastic Programming* (SP) methods are used to solve this optimization problem. The wait and see (WS) approach has been investigated in their works. Since VPCs can be regarded as abstractions of traffic flows, VP distribution becomes a flow-based optimization problem.

2.5 Review of Queueing Models

Queueing theory plays a key role in the modeling and analysis of networking problems [42]. A queueing system can be described as one having a service facility at which data packets arrive for service and where, whenever there are more packets in the system than the service facility can handle simultaneously, a queue is then formed. These packets take their turns for service according to a pre-assigned rule, and after service they leave the system.

Consider a buffer shown in Fig. 2-9. Data packets arrive and are buffered, ready to be read out on a transmission link at the rate of C bits/sec. The average arrival rate is denoted by λ packets/sec, and the service rate is denoted by μ packets/sec. In the more general queueing literature, the terms "jobs" or "customers" are used, rather than packets. A work-conserving queue is one in which packets must be transmitted or served, once admitted to the buffer, and in which the transmission link, the "server" in the queueing jargon, is never idle so long as there is at least one packet waiting for transmission [43]. We only consider work-conserving queues in this thesis.

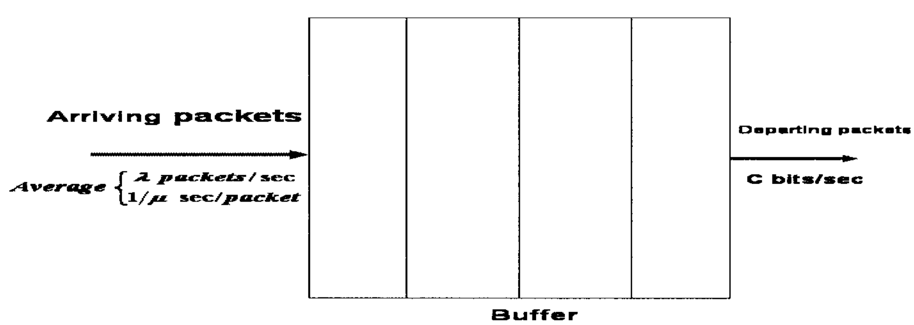


Figure 2-9. Model of buffering process.

Queueing theory enables us to determine the statistics of the queue, from which such desired performance parameters as the time spent waiting in the queue or the probabilities a packet is blocked or lost, on arrival, may be found. Generally, a queueing system is characterized as follows:

1. The packet arrival process: the specific arrival statistics of the incoming packets.
2. The packet-length distribution: comparable to the customer service time distribution when discussing customer arrivals in the queueing literature.
3. The number of servers and the service discipline: examples of the service discipline include *first-come, first-served* (FCFS) service, *last-in, first-out* (LIFO) service, and various types of priority service. Multiple output links or servers can be represented by the servers in parallel.

In order to abbreviate the description of a general queueing system, a convenient notation designed by D. G. Kendall has been universally used and standardized in the queueing literature. The Kendall notation in its most general form for an infinite queue is given as A/B/C, where "A" denotes the arrival distribution, "B" denotes the service time distribution and "C" stands for the number of servers used. The commonly used symbols for A and B are:

M : the exponential (Markovian) distribution;

D : the deterministic distribution;

E_k : the Erlang-k distribution;

G : a general distribution;

GI : a general distribution with independent inter-arrival or service times.

Hence, an M/G/1 queue would have an exponential inter-arrival distribution, a general service time distribution, and one server. An M/G/1/B queue is the same system with the further descriptor B denoting the finite buffer size (system capacity).

2.6 Introduction to Computational Complexity Theory

Complexity theory [44] is part of the theory of computation dealing with the resources required during computation to solve a given problem. The most common resources are *time* (how many steps needed to solve a problem) and *space* (how much memory needed to solve a problem). Other resources can also be considered, such as how many parallel processors are needed to solve a problem in parallel. Complexity theory differs from computability theory, which deals with whether a problem can be solved at all, regardless of the resources required.

A single "problem" is an entire set of related questions, where each question is a finite-length string. For example, the problem *FACTORIZE* is that, given an integer written in binary, return all of the prime factors of that number. A particular question is called an *instance*. For example, "give the factors of the number 15" is one instance of the *FACTORIZE* problem.

The time complexity of a problem is the number of steps that it takes to solve an instance of the problem, as a function of the size of the input (usually measured in bits) using the most efficient algorithm. For example, if an instance that is n bits long and can be solved in n^2 steps, then we say that it has a time complexity of n^2. Note that the exact number of steps will depend on exactly what machine or language is being used. To avoid this problem, we generally use the *Big O* notation. Big O notation is a type of symbolism used in complexity theory to describe the asymptotic behavior of functions. More exactly, it is used to describe an asymptotic upper bound for the magnitude of a function in terms of another usually simpler function. If a problem has time complexity $O(n^2)$ on one typical computer, then it will also have complexity $O(n^2)$ on most other computers. Hence this notation allows us to generalize away from the details of a particular computer.

Decision Problems

Much of complexity theory deals with *decision problems*. A decision problem is a problem where the answer is always YES/NO. For example, the problem *IS-PRIME* is that, given an integer written in binary, return whether it is a prime number or not. A decision problem is equivalent to a *language*, which is a set of finite-length strings. For a given decision problem, the equivalent language is the set of all strings for which the answer is YES.

Complexity Classes

Decision problems fall into sets of comparable complexity, called complexity classes. The complexity class *P* is the set of decision problems that can be solved by a deterministic machine in polynomial time. This class corresponds to an intuitive of idea of problems which can be effectively solved in the worst cases. The complexity class *NP* is the set of decision problems that can be solved by a non-deterministic machine in polynomial time. This class contains many problems which have the property that their solutions can be checked effectively. The question of whether P is the same set as NP is the most important open question in theoretical computer science.

Complexity theory often makes a distinction between YES answers and NO answers. For example, the set NP is defined as the set of problems where the YES instances can be checked "quickly" (i.e., in polynomial time). The set *Co-NP* is the set of problems where the NO instances can be checked quickly. The "Co" in the name stands for "complement". The *complement* of a problem is one where all the YES and NO answers are swapped.

Several well-known complexity classes are listed as follows:

P: solvable in polynomial time;

P-complete: the hardest problems in P to solve on parallel computers;

NP: YES answers checkable in polynomial time;

NP-complete: the hardest problems in NP;

Co-NP: NO answers checkable in polynomial time;

Co-NP-complete: the hardest problems in Co-NP.

2.7 Summary

In this chapter, we presented a brief review of different aspects of MPLS networks and the related topics, including the fundamentals of MPLS, QoS, TE, and the LSP optimization in the MPLS network. The queueing models and the computational complexity theory are presented briefly as well.

Chapter 3

Proposed Techniques

In the last chapter, we presented a brief review of background and related topics, such as fundamentals of the MPLS networks, QoS, MPLS traffic engineering, LSP optimization in MPLS, queueing models and computational complexity. In this chapter, we propose two new techniques for LSP optimization using TE in the MPLS networks.

The organization of this chapter is as follows. First, the classical network models are presented and analyzed. We then discuss a few ATM traffic flow assignment models as ATM virtual path and MPLS LSP have similar characteristics. We also briefly discuss an analytical LSR model associated with MPLS TE issues. The proposed analytical *Layer 2 cut-through switching* (L2-CTS) LSP model is presented in section 3.3.1. We then propose a nonlinear objective function to optimize the LSP traffic flow distribution in section 3.3.2.

3.1 Classical Network Models

In this section, we first present and analyze the classical capacity and flow assignment network models. More specifically, we discuss a few network modeling characteristics in the MPLS network, such as path-node incidence, load dependent parameters and multi-class traffic.

3.1.1 Generic Capacity and Flow Assignment Models

To analyze a communications network, switches or routers can be abstracted as nodes, while transmission lines, wired or wireless, can be abstracted as links. Accordingly, a communications network can be expressed by a graph G = (V, A), where V is the set of nodes (vertices) and A is the set of arcs (or links, edges). For example, a physical network can be defined as a graph, G = (V, A), where V is a finite nonempty set of nodes, and A is a collection of pairs of distinct nodes [45]. We note that A is indeed a set of all links in the network. In Fig. 3-1, a simple network is shown to illustrate the sets A, V and paths as follows:

V = {1, 2, 3, 4}

A = {(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)}

Paths: {1,2}, {1,3,2}, {1,3,4,2}, {1,4,2}, {1,4,3,2} are incident for OD pair (1,2)

Links: (1,3), (2,3) are incident on path {1,3,2}

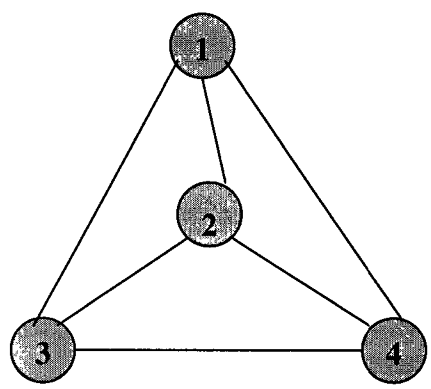


Figure 3-1. A prototype network.

Most packet communication networks can be regarded as an augmented graph with two types of indices: the operating index (e.g., packet delay, the number of packets in the system) denoted by T, and the capital index (e.g., capacities cost) denoted by D. In a communications network, two important entities must be considered: link flow λ_i and link transmission capacity C_i. Typically, λ_i corresponds to the traffic arrival rate on link i, and can be expressed as data units per second. The C_i has the same dimension as the λ_i. Accordingly, for a given network topology, the external traffic requirements, and the constraints of λ_i and C_i, three generic models are widely used in network analysis as follows:

Capacity Assignment (CA) model:

Given	λ_i
Minimize	T
Adjust	C_i
Constraint	D

Flow Assignment (FA) model:

Given	C_i
Minimize	T
Adjust	λ_i
Constraint	D

Capacity and Flow Assignment (CFA) model:

Minimize	T
Adjust	C_i and λ_i
Constraint	D

We note that the CA and FA models are special cases of the CFA model. These models can be used to minimize the operating index T (e.g., packet delay) subject to the upper bounded capital index D (e.g., capacity). However, based on different applications, the given parameters and the design variables are different.

3.1.2 Modeling Characteristics in MPLS Networks

The conventional network models presented in the last section are primarily characterized as follows:

- Link-node incidence;
- Load-independent parameters;
- Single class of traffic.

In the following, we introduce a few extended concepts that have been incorporated in current literature:

a) Path-node incidence

The formulations of the conventional network models are based on the link-node incidence. However, the link flows λ_i may further be expressed by the path flows x_k, then the network modeling is based on the path-node incidence [46,47]. Accordingly, the design variables are x_k and the model under this consideration becomes a *multi-commodity* flow (MCF) problem [48]. Note that each commodity in the MCF problem is composed of a source node s_i, a destination node t_i and a demand d_i. The objective of the MCF problem is to minimize the cost of routing a set of commodities simultaneously in the network, subjected to the capacity constraints.

b) Load dependent parameters and nonlinear models

From a mathematical point of view, all generic network models can be interpreted as the *shortest path* problem. The implementation of a generic model can be linear or nonlinear. If the network parameters are load independent (e.g., the cost per unit flow or the geographical length of links), then the resulting model usually takes the form of *linear programming* (LP). If the network parameters are load dependent (e.g., the mean or variance of delay), then the resulting model may take the form of *nonlinear programming* (NLP).

In modern packet switching techniques, there are two main paradigms: *datagram* and *virtual circuit*. For a datagram paradigm, such as IP networks, the use of nonlinear models is limited as they are likely to cause oscillations in terms of load distribution and congestion. The main reason is that the link parameters are updated much more frequently than the duration of a communicating session. For a virtual circuit paradigm, the situation is different [10]. Therefore, the nonlinear models are dominant in MPLS networks.

c) Multi-class traffic and path capacity

Different real-time requirements in multimedia communications result in multi-class traffic flows. These heterogeneous flows will traverse through all links associated with a particular *origin–destination* (OD) pair. Therefore, the conventional approach that treats the link entities as monolithic is no longer applicable. There is a need to further express link entities by path entities, where the traffic through each path may belong to a specific traffic class. Mathematically this is feasible because the total number of paths is usually greater than the total number of links.

By definition, transmission capacities in the CFA model are treated as design variables. The effect of link transmission capacities on the path flow assignment becomes ambiguous if the former is still used in the CFA model. It is more appropriate to use path capacities as design variables to replace link capacities. This approach is particularly natural for MPLS networking problems since the path capacity concept directly corresponds to the LSP provisioning.

3.2 Selected Models for Network Optimization Using TE

In recent years, several routing algorithms have been proposed for MPLS TE to establish bandwidth guaranteed paths and utilize existing network resources. A virtual traffic path optimization scheme with stochastic traffic in connection-oriented networks presented in [4] distributes the traffic flows through all available paths evenly using constraint-based routing, taking the redundant path capacities into account. A dynamic online routing algorithm presented in [5] allocates paths with reserved bandwidth evenly across the network to balance the traffic load and minimize network congestions. A stochastic offline traffic engineering framework presented in [6] optimizes the bandwidth provisioning and the path selection in terms of maximizing revenue from serving demands, which are uncertain and specified by probability distributions. In addition, minimum interference routing algorithm [7], variations of *Open Shortest Path First* (OSPF) routing schemes [8], and static routing algorithm [9] have also been used in the MPLS TE applications.

Meanwhile, several works have been done to efficiently combine Layer 2 switching and Layer 3 routing for fast IP packets transmission using TE in the MPLS network. An LSR model presented in [3] focuses on the IP datagram transmission delay in both Layer 2 and Layer 3. A Layer 3 switching mechanism presented in [49] analyzes the datagram waiting time and the out-of-order transmission rate of datagrams.

Because ATM virtual path and MPLS LSP have similar characteristics, we present an ATM traffic flow assignment model [4,10] in section 3.2.1. We then briefly present an analytical LSR model [3] associated with MPLS TE issues in section 3.2.2.

3.2.1 Generic FA Model in the ATM Network

The FA model presented in [4,10] is typically used in ATM VP distribution optimization. Here, a communications network is described by two entities: the link flow (i.e., the traffic arrival rate on the link), and the link transmission capacity. The objective function in the FA model [10] is defined as follows:

Minimize $$Y = \sum_{w \in W} \sum_{p \in P_w} D_p(x_p) \qquad (3\text{-}1)$$

Subject to $$\sum_{p \in P_w} x_p = \gamma_w \qquad (3\text{-}2a)$$

$$\sum_{p \in Q_i} x_p \le C_i \qquad (3\text{-}2b)$$

$$x_p \ge 0 \; ; \forall p \in P_w, w \in W, i \in L$$

where the symbols are defined as follows:

x_p: the flow of path p (the design variable).

D_p: the flow cost function of path p.

W: the set of all origin and destination (OD) pairs.

P_w: the set of all paths that connect a particular OD pair w.

γ_w: the input flow for OD pair w.

C_i: the capacity of link i.

Q_i: the set of all paths that pass link i.

L: the set of all links.

The FA model is based on the link-node incidence, which is suitable for conventional data networks. The model describes the network based on link-node relations. In Eq. (3-2b), the left side of the equation is the link flow expressed as path flows. The objective function minimizes the network operating index represented by Y in Eq. (3-1). The generalized cost function D_p in Eq. (3-1) can be expressed in different forms, such as physical distance, unit traffic flow cost, utilization ratio, and the degree of congestion. By extending D_p to different forms, we can get variants of the FA model, and different optimization objectives can be achieved. Hence, we refer to this FA model as the *generic flow assignment* (GFA) model. For conventional data networks, connectionless service is dominant, so the network can be described by the relations between the links and the nodes. Therefore, the above link-node incidence based on the GFA model is suited in conventional data networks.

In ATM networks, the above GFA model is generally used for VP optimization. For example, a variant of the FA model is used in [45] by extending the FA model to include additional features. A flow-based optimization model is also used in [39]. In fact, this flow-based optimization model is a variant of the GFA model.

For MPLS networks, connection-oriented traffic is dominant, and hence the link-node incidence may not be suitable for the network modeling. The GFA model needs to be modified for MPLS network modeling.

3.2.2 LSR-NTKO Model

Nakazawa, Tamura, Kawahara, and Oie [3] presented an analytical LSR model (henceforth referred to as the LSR-NTKO model) to analyze the performance of IP datagram transmission delay. The model combines Layer 3 forwarding with Layer 2 high-speed switching, and transfers IP datagrams fast by cut-through transmission via LSPs on Layer 2. Note that there are two cases of IP datagram transmissions in the LSR: hop-by-hop transmission via Layer 3 routing kernel, and cut-through transmission via Layer 2 LSPs.

In the LSR-NTKO model, the queueing model used in Layer 2 is the batch_geom/D/1/K, with the arrival rate λ and the batch size N (sources). The Layer 3 routing kernel was modeled by the batch_geom/D/1 queue, with the service rate $1-\omega$. Note that ω is the ratio of the cut-through transmission bandwidth and the physical circuits bandwidth. The average datagram transmission delay in the Layer 3 routing kernel was shown to be:

$$W_{def} = \frac{1}{\lambda(1-R_C)} \sum_{b=1}^{\infty} (b \sum_{i=0}^{N_p} x(b,i)) \tag{3-3}$$

where the cut-through rate R_C is the ratio of the average number of datagrams transmitted by cut-through LSPs and the average number of datagrams generated from all sources when the maximum number of cut-through LSP is set to N_p. $x(b,i)$ is the system steady state probability, in which b is the queue buffer length of the routing kernel, and i is the number of the cut-through LSPs.

On the other hand, the average datagram transmission delay in the Layer 2 switching by cut-through LSPs was shown to be:

$$W_{cut} = \frac{1}{\lambda R_C} \sum_{b=0}^{\infty} \sum_{i=1}^{N_p} i x(b,i) \tag{3-4}$$

Thus, the total average datagram processing delay in an MPLS core network can be obtained by summing the delays both on Layer 2 and Layer 3 as follows:

$$W = (1-R_C) W_{def} + R_C W_{cut} \tag{3-5}$$

3.3 Proposed Techniques

The proposed analytical *Layer 2 cut-through switching* (L2-CTS) LSP model is presented in section 3.3.1. The LSP traffic flow distribution optimization objective function is then presented in section 3.3.2.

3.3.1 Proposed Analytical L2-CTS LSP Model

It is common knowledge that different kinds of LSP transmission mechanisms can be set up in the MPLS network [1]. We can treat the TE issue by setting up different types of LSPs through the MPLS network. In this section, we propose an analytical L2-CTS LSP model, shown in Fig. 3-2, in which the LSPs are divided into two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing.

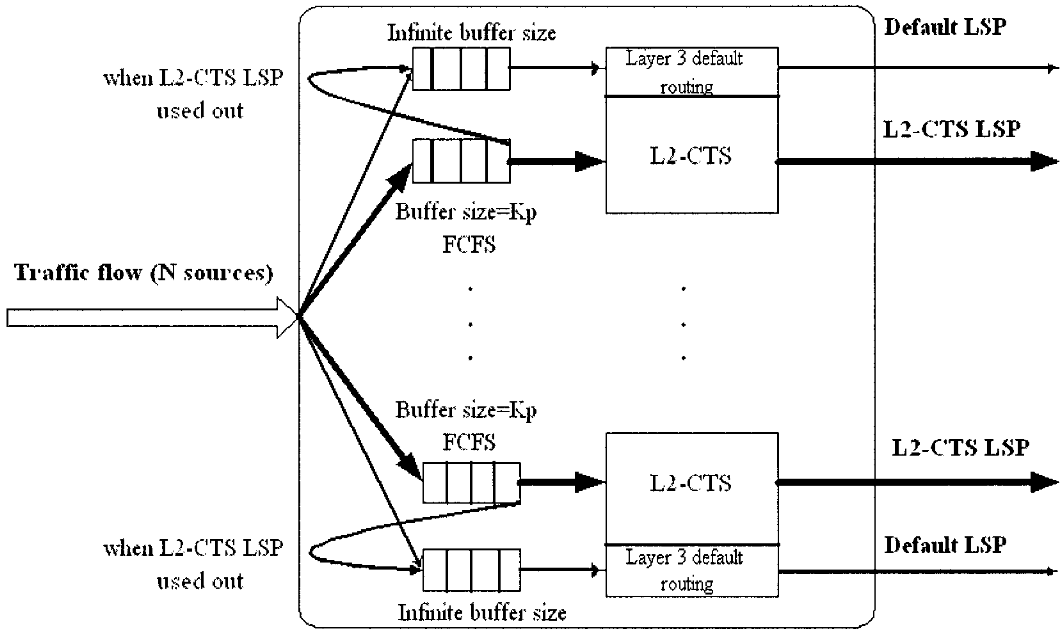


Figure 3-2. Proposed analytical L2-CTS LSP model.

Note that the objectives of our proposed L2-CTS LSP model and the LSR-NTKO model [3] are different. The LSR-NTKO model focuses only on the processing delay in each LSR, and it can be used to reduce the datagram transmission delay on each hop of the network (i.e., each LSR in the MPLS domain). On the other hand, the objective of the proposed L2-CTS model is to achieve the overall minimized transmission weighted delay and more even traffic flow distribution on each link of the network.

a) Transmission mechanism

In our proposed analytical L2-CTS LSP model, we assume the label mapping method as a data-driven (flow-driven) scheme [1]. At the beginning of each transmission, IP flows are raised to the Layer 3 routing kernel and transmitted by the default LSPs which are pre-established for hop-by-hop transmission. Once L2-CTS LSPs are set up, all IP flows are transmitted by the L2-CTS LSPs until all available network resources assigned on Layer 2 are utilized. Only when all available L2-CTS LSPs are used out, the overflowed traffic will be raised to the Layer 3 transmission. Hence, the connection-oriented forwarding characteristics of Layer 2 switching are achieved while retaining the equally desirable flexibility and scalability of Layer 3 routing.

b) Assumptions and validations

In order to validate the proposed L2-CTS LSP model, some important modeling assumptions and their justifications are presented below:

Various traffic models [43,50,51] (e.g., *Markov Modulated Poisson Process* (MMPP), self-similar arrivals) have been used for real-time traffic flow analysis for individual or multiplexed sources. In order to make the analysis simpler, we choose the Poisson arrival distribution in this thesis. For the same reason, we assume that the service times are exponentially distributed. For OD pair w, each L2-CTS LSP is queued separately with a virtual buffer size k_p, and the service discipline is first come, first served (FCFS).

In summary, for OD pair w, the virtual queue system assumptions [52] shown in Table 3-1 are as follows:

- The time between successive arrivals is exponentially distributed;
- The service times are exponentially distributed;
- There is one server (head end LSR);
- Each L2-CTS LSP is queued separately with a virtual buffer size k_p, while the buffer size for Layer 3 default routing is infinite;
- The service discipline is first come, first served.

Table 3-1. Summary of assumptions. $E[s]$ is the mean service time per job.

	Layer 2 cut-through switching LSPs	Layer 3 default routing LSPs
Queue Model	$M/M/1/k_p$	M/M/1
Arrive Rate	λ	λP_B
Service Rate	1/E[s]	1-1/E[s]
Buffer Size	k_p	infinite

For the M/M/1/B queueing [52], the mean response time $E[r]$ can be calculated using the following equation (r is the response time and the mean response time is then denoted by $E[r]$. The same notations for $E[s]$ and $E[n]$):

$$E[r] = \frac{E[n]}{\lambda(1 - P_B)}$$

where λ is the average arrival rate, $E[n]$ is the mean number of jobs in the system, and P_B is the probability of having B jobs simultaneously in the system. The P_B can be calculated as follows:

$$P_B = \begin{cases} \dfrac{1-\rho}{1-\rho^{B+1}}\rho^B & otherwise \\ \dfrac{1}{B+1} & \rho = 1 \end{cases}$$

where ρ ($= \lambda E[s]$) is the traffic intensity.

From Little's law [52], we know:

Mean number of jobs in the system = Mean throughput × Mean response time

That is,

$$E[n] = mean \quad throughput \quad \times E[r]$$

If the average arrival rate for each L2-CTS LSP is assumed to be λ, the cut-through bandwidth B_{CT} (which is the mean throughput of Layer 2 transmission) of each L2-CTS LSP can be calculated as:

$$B_{CT} = \frac{E[n]}{E[r]} = \lambda(1 - P_B)$$

Using the above relationship, the cut-through switching rate for each L2-CTS LSP can be calculated as:

$$R_{L2-CTS} = \begin{cases} \dfrac{k_p}{k_p + 1} & \rho_p = 1 \\ \dfrac{1-\rho_p^{k_p}}{1-\rho_p^{k_p+1}} & otherwise \end{cases} \tag{3-6}$$

where $\rho_p = \lambda E[s]$, which is the traffic intensity for each L2-CTS LSP.

Eq. (3-6) provides the L2-CTS rate of the proposed model. It can be used to modify the GFA model to distribute the traffic flow more evenly through different LSPs on Layer 2.

3.3.2 Proposed LSP Traffic Flow Distribution Optimization Objective Function

To reduce the overall operations cost, we need to prevent the situation where some parts of the network are over-utilized (congested), while other parts are under-utilized. As a result, the optimization of traffic flow distribution on a given network is an important goal of the TE. The GFA model [10] has been widely used to optimize the distribution of virtual paths in ATM networks. In this section, we apply the GFA model to obtain a new LSP traffic flow distribution optimization objective function. This objective function minimizes the overall weighted delay for the whole network transmission and distributes the traffic flow more evenly across different LSPs.

Because the traffic flows are mostly transmitted by the L2-CTS LSPs in the data-driven scheme, we only consider L2-CTS LSPs transmission for the performance optimization in the MPLS domain. The Layer 3 default routing transmission happens usually at the beginning of the new in-coming traffic flow. After the setting up of L2-CTS LSPs, the traffic flows will be mostly through the Layer 2 LSPs if the network topology is not changed. Only when all available L2-CTS LSPs are used out, the overflowed traffic will be raised to the Layer 3 transmission. (more discussion follows in section 4.1)

The proposed optimization mechanism can be established in the head end LSRs by setting up and distributing LSPs, and it can be updated dynamically with changes to the topology.

In order to develop the optimization objective function, we define a few symbols, in addition to the symbols defined in Eq. (3-1) and (3-2), as follows:

d_p: the unit flow cost of path p.

N_p: the capacity of path p.

k_p: the virtual queue buffer size for each L2-CTS LSP (i.e., the maximum available network resources that can be assigned on each L2-CTS LSP for OD pair w).

ρ_p: the traffic intensity for each L2-CTS LSP.

β: the specified packet discard probability.

According to the objective function (3-1) in the GFA model, the generalized cost function D_p in Eq. (3-1) can be expressed in different forms, such as physical distance, unit traffic flow cost, utilization ratio, and the degree of congestion. In the proposed objective function, we use d_p, the unit flow cost of path p, in place of D_p. In addition, we only consider the Layer 2 transmission as explained above, and hence we use the R_{L2-CTS} as given by Eq. (3-6) to develop the objective function. The proposed objective function for optimizing the traffic flow distribution through the L2-CTS LSPs is as follows:

Minimize $$Z = \sum_{w \in W} \sum_{p \in P_w} d_p \frac{1-\rho_p{}^{k_p}}{1-\rho_p{}^{k_p+1}} (x_p) \tag{3-7}$$

Subject to $$\sum_{p \in P_w} x_p = \gamma_w \tag{3-8a}$$

$$\sum_{p \in Q_i} N_p \le C_i \tag{3-8b}$$

$$\frac{x_p}{N_p} \le \frac{2k_p}{2k_p - \ln(\beta)} \tag{3-8c}$$

$$x_p \ge 0 \text{ , } N_p \ge 0 \text{ , } k_p \ge 1$$

$$\forall p \in P_w, w \in W, i \in L$$

where x_p, N_p and k_p are the design variables.

The objective function minimizes the overall weighted delay for the whole network transmission represented by Z in Eq. (3-7). Note that Eq. (3-8a) ensures that the assigned path flows between OD pair w satisfy the input flows. On the other hand, Eq. (3-8b) ensures that the total path capacities through link i should not exceed the capacity of link i. Finally, Eq. (3-8c) ensures that the utilization of path p does not exceed the upper bound of congestion for path p (UBCP). It was shown by Liu [11] that the UBCP can be approximately expressed as follows:

$$UBCP = \frac{2k_p}{2k_p - \ln(\beta)}$$

The unit flow cost d_p for each path can be derived from the unit link flow cost. In terms of weighted delay, the unit link flow cost is primarily determined by the physical length of each link. Hence, we can express d_p as the physical length of path p between OD pair w.

Note that in the GFA model, the network modeling concept is based on the link-node incidence. In MPLS networks, however, it is more appropriate to use the concept of path-node incidence in the LSP traffic flow distribution. There are two main concepts here: path flow and path capacity. Path flows are optimized to meet external traffic demands while path capacities are optimized to occupy all physical link capacities. By introducing these two concepts, the proposed LSP flow distribution optimization becomes a CFA model on the logical network level, but on the physical network level, it still remains an FA model. It is natural to use path entities to replace link entities, where the traffic flow through each LSP may belong to a specific traffic class according to multi-class traffic flows in MPLS networks.

It is obvious that the proposed LSP optimization objective function (3-7) is nonlinear which is based on load dependant parameters, because it is to minimize the overall network transmission weighted delay. This is feasible since the traffic oscillation in conventional data networks is no longer dominant in MPLS networks. For a quasi-static topology network, the nonlinear model usually can handle the congestion better than the linear model [11].

We achieve two main objectives with (3-7). First, the overall weighted transmission delay is minimized, and traffic flows among different links are more evenly distributed. This is expected to make the network more robust to the physical link failure. Secondly, the objective function (3-7) provides a set of optimum path capacity assignment N_p for the logical network design. It also provides the optimum k_p , which indicates the optimized available network resources that can be assigned on L2-CTS LSPs. Thus, we can balance the resource utilization and decrease the possibility of congestions.

3.4 Summary

In this chapter, we have proposed two techniques for LSP optimization using TE in MPLS networks, including the analytical L2-CTS LSP model and the LSP traffic flow distribution optimization objective function. We treat the TE issue by setting up different types of LSPs through the MPLS network. The proposed L2-CTS LSP model assumes two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing. Using this model, we can achieve Layer 2 simplified connection-oriented forwarding while retaining Layer 3 routing flexibly and scalably. The incorporated LSP traffic flow distribution optimization objective function is to minimize the overall network transmission weighted delay and optimize the traffic flow distribution across different LSPs.

The L2-CTS LSP model is based on the path-node incidence, and hence is suitable for the dominant multi-class traffic flows in MPLS networks. The nonlinear LSP optimization objective function can balance the resource utilization, distribute traffic flows more evenly and decrease the possibility of congestions.

Chapter 4

Numerical Results and Discussions

In the last chapter, we proposed two new techniques for LSP optimization using TE in MPLS networks. First, we proposed an analytical L2-CTS LSP model, in which LSPs are divided into two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing. Using this model, the connection-oriented forwarding characteristics of Layer 2 switching are achieved while retaining the equally desirable flexibility and scalability of Layer 3 routing. Secondly, we proposed the incorporated LSP traffic flow distribution optimization objective function to minimize the overall network transmission weighted delay and optimize the traffic flow distribution across different LSPs.

In this chapter, we will first analyze the Layer 2 cut-through transmission rate of the proposed L2-CTS LSP model mathematically. We will then evaluate the performance of the LSP system resulting from the L2-CTS LSP model and the incorporated LSP flow distribution objective function, and compare it with the conventional virtual path optimization model in two different prototype networks. Furthermore, we will characterize the numerical results and analyze the computation complexity of our proposed techniques for LSP optimization in MPLS networks.

4.1 Layer 2 Cut-through Transmission Rate Analysis

In this section, we present a brief mathematical analysis of the R_{L2-CTS} of the proposed L2-CTS LSP model used for LSP optimization in MPLS networks.

Eq. (3-6) relates the R_{L2-CTS} with the virtual buffer size k_p and the traffic intensity ρ. Fig. 4-1 shows the variation of R_{L2-CTS} for different k_p and ρ. When $\rho \leq 1$, it is observed that, the R_{L2-CTS} increases sharply with k_p when $k_p < 6$. However, when $k_p \geq 6$, the R_{L2-CTS} becomes saturated and increases slowly. In other words, if the buffer size k_p (i.e., the available network resources that can be assigned on L2-CTS LSPs) is large, the R_{L2-CTS} approaches one (it can be verified by Eq. (3-6), that is, R_{L2-CTS} converges to 1 as $k_p \to \infty$). This is because in the data-driven scheme, the Layer 3 default routing transmission usually happens at the beginning of each transmission if the available network resources are enough to be assigned on the L2-CTS transmission. After several L2-CTS LSPs are set up, the traffic flow will be mostly through the Layer 2 cut-through switching. Under this circumstance, Layer 3 default routing transmission can be ignored in the later simulation, in which we will evaluate the LSP system resulting from the proposed L2-CTS LSP model.

On the other hand, when $\rho > 1$, the R_{L2-CTS} decreases. From Eq. (3-6), it can be observed that the R_{L2-CTS} converges to $\frac{1}{\rho}$ as $k_p \to \infty$. Therefore, the traffic flow through the conventional Layer 3 routing will increase as all L2-CTS LSPs are used out. For these traffic flows through the Layer 3 routing, we can still use the conventional hop-by-hop transmission, and hence it is unlikely that the overflowed traffic through Layer 3 will have any negative impact on the performance of the LSP system resulting from the proposed L2-CTS model. Furthermore, for the same number of k_p, if the traffic intensity increases, the traffic flow through the Layer 2 switching will decrease.

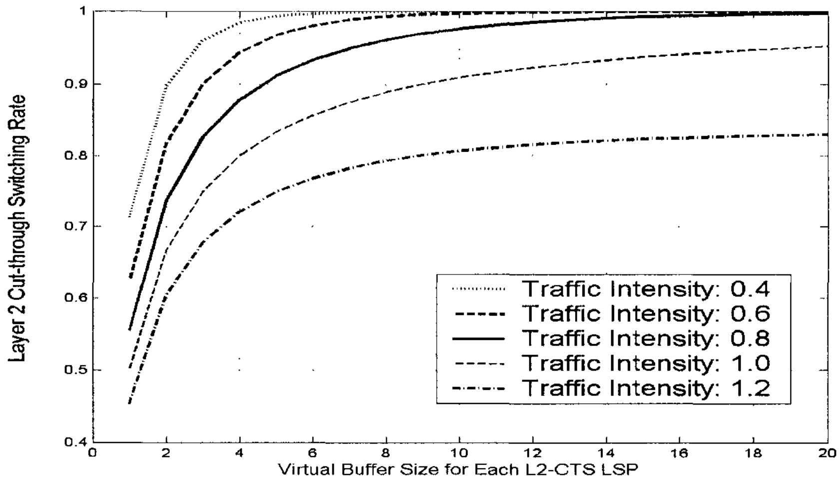


Figure 4-1. Layer 2 cut-through switching (L2-CTS) rate vs. the virtual queue buffer size k_p for each L2-CTS LSP.

The geometric perspective of R_{L2-CTS} associated with the virtual queue buffer size k_p and the traffic intensity for each L2-CTS LSP is shown in Fig. 4-2.

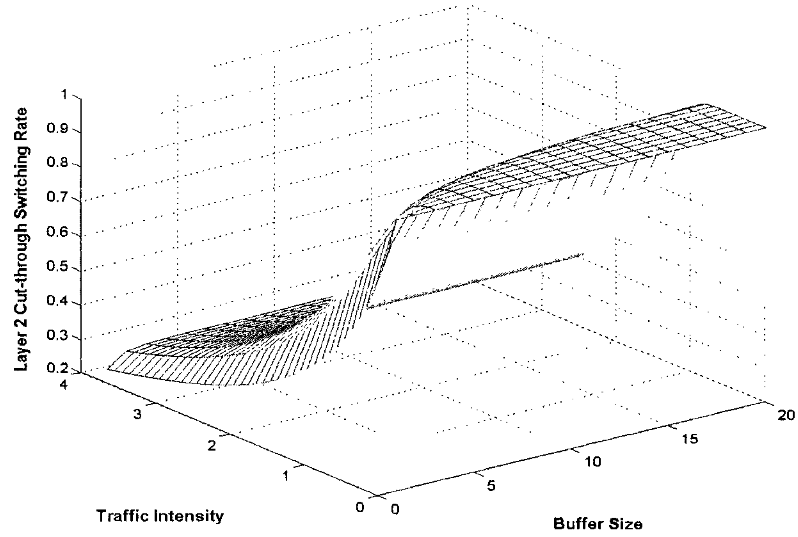


Figure 4-2. Distribution of Layer 2 cut-through switching (L2-CTS) rate.

4.2 Performance Evaluation of LSP Systems

In this section, we evaluate the performance of the LSP system resulting from the proposed L2-CTS LSP model and the incorporated LSP flow distribution objective function (henceforth referred to as the LSP-L2), and compare it with the LSP system resulting from a *conventional virtual path* (CVP) optimization model [1,10] (henceforth referred to as the LSP-CVP).

The LSP-CVP is based on the link-node incidence and treats the traffic flow distribution as the link capacity assignment, whereas the LSP-L2 is based on the path-node incidence.

The performance of the LSP-CVP has been evaluated with the GFA objective function (3-1), whereas the LSP-L2 has been evaluated with the modified objective function (3-7). The optimization for the two systems has been carried out using the *General Algebraic Modeling System* (GAMS) [53]. Note that the LSP-L2 is a nonlinear model, and the corresponding optimization needs *nonlinear programming* (NLP) methods. GAMS is a high-level modeling system for mathematical programming problems, and it is capable of solving large-scale linear and nonlinear programming models (see Appendix A). GAMS consists of a language compiler and a stable of integrated high-performance solvers. For NLP, the GAMS can not guarantee that we can find the globally optimal solution. For most cases, we can only get the locally optimal solution.

We use two different prototype networks for the performance evaluation. A medium size network is presented first in section 4.2.1, and it is followed by a more complex network simulation in section 4.2.2.

To evaluate the performance, the flow cost function D_p in the LSP-CVP takes the form of d_p (the unit flow cost of path p), which is the same d_p used in the LSP-L2. Therefore, the objective function of the LSP-CVP is expressed in terms of the total delay in the network.

We use the *link utilization ratio* (LUR) as the indicator of the link load in the simulation. In this thesis, the LUR of link i is defined as follows:

$$LUR_i = \frac{\text{Total traffic flow through link } i}{\text{The capacity of link } i}$$

Correspondingly, the *standard deviation of link utilization ratio* (SDLUR) is used as the performance criteria to indicate the spreading of traffic load in the network, and the SDLUR is defined as follows:

$$SDLUR = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(LUR_i - \mu)^2}$$

where L is the total number of links, and μ is the mean value of the LUR, which is defined as follows:

$$\mu = \frac{1}{L}\sum_{i=1}^{L} LUR_i$$

Note that the lower the standard deviation, the higher evenness of the traffic distribution we can achieve. For the reason we mentioned in section 4.1, we only consider the L2-CTS transmission for each OD pair in the simulation.

4.2.1 Performance Evaluation in Prototype Network-1

In order to evaluate the performance, we firstly apply both the LSP-CVP and the LSP-L2 on a medium size prototype network-1, which has 12 nodes and 17 links as shown in Fig. 4-3, to simulate the distribution of L2-CTS LSPs. For each link, the corresponding link capacity and link distance are shown in the figure. The objective of this simulation is to assign the optimal capacities for different L2-CTS LSPs. In each experiment, only 10 random OD pairs are considered, and we assume that for each OD pair, there are at most two LSPs. These two LSPs are the two shortest disjointed paths to connect the OD pair. We use Dijkstra's algorithm [48] to find these two LSPs for each OD pair (see Appendix B). Hence, there are at most 20 paths under consideration for each scenario. After simulation, each path is assigned a capacity. A capacity of zero corresponds to a non-existent path.

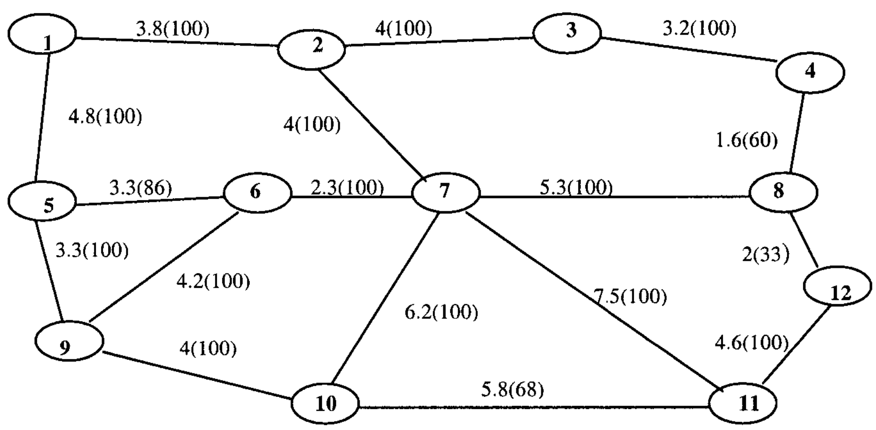


Figure 4-3. A prototype network with 12 nodes. The numbers over each link denote distance (capacity).

Both the LSP-L2 and the LSP-CVP are applied to this prototype network, and the optimization is performed for both models using GAMS. For the LSP-CVP, it is an LP problem, whereas for the LSP-L2, it is an NLP problem. The main steps of the experiment are as follows:

1. Choose 10 OD pairs randomly and use Dijkstra's algorithm to find out the 20 shortest disjointed paths.

2. For the chosen 10 OD pairs, use a random number generator to generate the set of 10 external traffic demands γ_{ω}. The γ_{ω}'s are created with uniform distribution with values in the range [30, 90] requests/ time-unit.
3. For each OD pair, assume $\rho = 0.7$, $N = 20$ and $\beta = 0.001$.
4. Use GAMS to perform the optimization.
5. For each OD pair w, the arrival traffic flow follows a Poisson distribution. Use a Poisson random number generator to generate 20 sets of numbers with mean value that equals to the corresponding external traffic flow number that was generated in Step 2. Each set of Poission random numbers consists of one simulation scenario.
6. Repeat Step 4.

Table 4-1 shows all paths and routes associated with OD pairs and the corresponding path costs.

Table 4-1. OD pairs, paths, routes and their costs.

OD Pair Index	OD Pair	Path Index	Routes and Nodes	Path Cost (Distance)
1	(1, 7)	P1	{1, 2, 7}	7.8
		P2	{1, 5, 6, 7}	10.4
2	(2, 5)	P3	{2, 1, 5}	8.6
		P4	{2, 7, 6, 5}	9.6
3	(3, 10)	P5	{3, 2, 7, 10}	14.2
		P6	{3, 4, 8, 12, 11, 10}	17.2
4	(4, 2)	P7	{4, 3, 2}	7.2
		P8	{4, 8, 7, 2}	10.9
5	(5, 10)	P9	{5, 9, 10}	7.3
		P10	{5, 6, 7, 10}	11.8
6	(6, 11)	P11	{6, 7, 11}	9.8
		P12	{6, 9, 10, 11}	14
7	(7, 9)	P13	{7, 6, 9}	6.5
		P14	{7, 10, 9}	10.2
8	(8, 11)	P15	{8, 12, 11}	6.6
		P16	{8, 7, 11}	12.8
9	(9, 1)	P17	{9, 5, 1}	8.1
		P18	{9, 6, 7, 2, 1}	14.3
10	(12, 10)	P19	{12, 11, 10}	10.4
		P20	{12, 8, 7, 10}	13.5

Table 4-2. Links and paths incidence. "*" is used to indicate the corresponding path passing the link.

Path \ Link	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P1	*		*														
P2		*		*	*												
P3	*	*															
P4			*	*	*												
P5			*						*	*							
P6											*	*		*	*	*	
P7										*	*						
P8			*									*	*				
P9						*		*									
P10				*	*				*								
P11					*												*
P12							*	*								*	
P13					*		*										
P14								*	*								
P15														*	*		
P16													*				*
P17		*				*											
P18	*		*		*		*										
P19															*	*	
P20									*				*	*			

Table 4-3. Network parameters used in the simulation. C_i is the capacity of link i, and γ_w is the input flow for OD pair w.

C_i, $i = 1,\dots,17$			γ_w, $w = 1,\dots,10$	
C_1	100		γ_1	32
C_2	100		γ_2	56
C_3	100		γ_3	42
C_4	86		γ_4	60
C_5	100		γ_5	53
C_6	100		γ_6	72
C_7	100		γ_7	51
C_8	100		γ_8	49
C_9	100		γ_9	37
C_{10}	100		γ_{10}	65
C_{11}	100			
C_{12}	60			
C_{13}	100			
C_{14}	33			
C_{15}	100			
C_{16}	68			
C_{17}	100			

Table 4-2 shows the path and link incidence. Table 4-3 shows some network parameters that we are using in the simulation. In this thesis, we have performed the optimization for both the LSP-CVP and the LSP-L2 using the GAMS.

4.2.1.1 Performance of the LSP-CVP

The objective function for the LSP-CVP optimization is given by Eq. (3-1). In order to perform this optimization, we change the limits of the sum in Eq. (3-1) according to the parameters of the prototype network as shown in Fig. 4-3. Thus, the reformulated objective function of the LSP-CVP is as follows:

$$\text{Minimize} \quad Y = \sum_{p=1}^{20} d_p x_p \tag{4-1}$$

Subject to the following constraints:

Constraints obtained from Eq. (3-2a):

$$x_1 + x_2 = \gamma_1$$

$$x_3 + x_4 = \gamma_2$$

$$x_5 + x_6 = \gamma_3$$

$$x_7 + x_8 = \gamma_4$$

$$x_9 + x_{10} = \gamma_5$$

$$x_{11} + x_{12} = \gamma_6$$

$$x_{13} + x_{14} = \gamma_7$$

$$x_{15} + x_{16} = \gamma_8$$

$$x_{17} + x_{18} = \gamma_9$$

$$x_{19} + x_{20} = \gamma_{10}$$

Constraints obtained from Eq. (3-2b):

$$x_1 + x_3 + x_{18} \le C_1$$

$$x_2 + x_3 + x_{17} \le C_2$$

$$x_1 + x_4 + x_5 + x_8 + x_{18} \le C_3$$

$$x_2 + x_4 + x_{10} \le C_4$$

$$x_2 + x_4 + x_{10} + x_{11} + x_{13} + x_{18} \le C_5$$

$$x_9 + x_{17} \le C_6$$

$$x_{12} + x_{13} + x_{18} \le C_7$$

$$x_9 + x_{12} + x_{14} \le C_8$$

$$x_5 + x_{10} + x_{14} + x_{20} \le C_9$$

$$x_5 + x_7 \le C_{10}$$

$$x_6 + x_7 \le C_{11}$$

$$x_6 + x_8 \le C_{12}$$

$$x_8 + x_{16} + x_{20} \le C_{13}$$

$$x_6 + x_{15} + x_{20} \le C_{14}$$

$$x_6 + x_{15} + x_{19} \le C_{15}$$

$$x_6 + x_{12} + x_{19} \le C_{16}$$

$$x_{11} + x_{16} \le C_{17}$$

and $x_p \ge 0$, $p = 1,\ldots,20$.

Table 4-4. Optimized path flows for the LSP-CVP in scenario 1.

x_1	36.00		x_{11}	78.00
x_2	0		x_{12}	0
x_3	61.00		x_{13}	22.00
x_4	0		x_{14}	32.00
x_5	46.00		x_{15}	31.00
x_6	0		x_{16}	22.00
x_7	54.00		x_{17}	36.00
x_8	11.00		x_{18}	0
x_9	58.00		x_{19}	68.00
x_{10}	0		x_{20}	2.00

After performing the optimization using GAMS, the result for an arbitrary scenario is shown in Table 4-4. It shows the assigned traffic flows for each path in the LSP-CVP.

The minimized overall transmission delay Y = 5136.50 (as calculated by Eq. (4-1), with d_p as shown in Table 4-1).

When the CVP model is used for LSP optimization, the optimized traffic flows are considered as the optimal LSP capacity assignment. For instance, it is observed that the flow in Path 2 is zero. That means, Path 2 does not exist and all the traffic for OD pair (1,7) is carried by Path 1. Because path capacities and path flows are considered to be the same, there is no redundant capacity assigned to accommodate the external traffic changes. Consequently, the resulting LSP system is not resilient to traffic changes. In addition, we can observe that the traffic is not distributed evenly in the network because of the nature of the shortest path algorithm. We will discuss more details about this in section 4.3.

4.2.1.2 Performance of the LSP-L2

The objective function for the LSP-L2 optimization is given by Eq. (3-7). In order to perform this optimization, we change the limits of the sum in Eq. (3-7) according to the parameters of the prototype network as shown in Fig. 4-3. Thus, the reformulated objective function of the LSP-L2 is as follows:

$$\text{Minimize} \qquad Z = \sum_{p=1}^{20} d_p \frac{1-\rho^{k_p}}{1-\rho^{k_p+1}}(x_p) \qquad (4\text{-}2)$$

Subject to the following constraints:

<u>Constraints obtained from Eq. (3-8a):</u>

$x_1 + x_2 = \gamma_1$

$x_3 + x_4 = \gamma_2$

$x_5 + x_6 = \gamma_3$

$x_7 + x_8 = \gamma_4$

$x_9 + x_{10} = \gamma_5$

$x_{11} + x_{12} = \gamma_6$

$x_{13} + x_{14} = \gamma_7$

$x_{15} + x_{16} = \gamma_8$

$x_{17} + x_{18} = \gamma_9$

$x_{19} + x_{20} = \gamma_{10}$

<u>Constraints obtained from Eq. (3-8b):</u>

$N_1 + N_3 + N_{18} \le C_1$

$N_2 + N_3 + N_{17} \le C_2$

$N_1 + N_4 + N_5 + N_8 + N_{18} \le C_3$

$N_2 + N_4 + N_{10} \le C_4$

$N_2 + N_4 + N_{10} + N_{11} + N_{13} + N_{18} \le C_5$

$N_9 + N_{17} \le C_6$

$N_{12} + N_{13} + N_{18} \le C_7$

$N_9 + N_{12} + N_{14} \le C_8$

$N_5 + N_{10} + N_{14} + N_{20} \le C_9$

$N_5 + N_7 \le C_{10}$

$N_6 + N_7 \le C_{11}$

$N_6 + N_8 \le C_{12}$

$N_8 + N_{16} + N_{20} \le C_{13}$

$N_6 + N_{15} + N_{20} \le C_{14}$

$N_6 + N_{15} + N_{19} \le C_{15}$

$N_6 + N_{12} + N_{19} \le C_{16}$

$N_{11} + N_{16} \le C_{17}$

<u>Constraints obtained from Eq. (3-8c):</u>

$$\frac{x_p}{N_p} \leq \frac{2k_p}{2k_p - \ln(\beta)}, \ p = 1,...,20.$$

and $x_p \geq 0$, $N_p \geq 0$, $k_p \geq 0$, $p = 1,...,20$.

After performing the optimization using GAMS, the result for an arbitrary scenario is shown in Table 4-5. It shows the assigned flow and capacity for each path in the LSP-L2.

Table 4-5. Optimized path flow and capacity assignments for the LSP-L2 in scenario 1.

Path Flow			Path Capacity	
x_1	29.48		N_1	34.71
x_2	6.52		N_2	7.68
x_3	55.45		N_3	65.29
x_4	5.55		N_4	6.53
x_5	34.92		N_5	41.11
x_6	11.08		N_6	13.05
x_7	50.01		N_7	58.89
x_8	14.99		N_8	17.65
x_9	51.15		N_9	69.12
x_{10}	6.85		N_{10}	8.06
x_{11}	78.00		N_{11}	81.82
x_{12}	0		N_{12}	0
x_{13}	34.08		N_{13}	40.13
x_{14}	19.92		N_{14}	30.88
x_{15}	53.00		N_{15}	62.41
x_{16}	0		N_{16}	0
x_{17}	36.00		N_{17}	42.39
x_{18}	0		N_{18}	0
x_{19}	53.06		N_{19}	61.13
x_{20}	16.94		N_{20}	19.95

The minimized overall transmission delay of the LSP-L2, $Z = 4796.29$ (as calculated by Eq. (4-2), with d_p as shown in Table 4-1), whereas the minimized overall transmission delay of the LSP-CVP is 5136.50.

It is observed that the path capacity assigned for each path is always higher than the path flow (except for those paths where the path flow is zero). In other words, in the LSP-

L2, each path is assigned some redundant capacities and the external traffic demands are satisfied by the path flows assignments. Hence, the LSP-L2 is resilient to traffic changes. It contrasts with the LSP-CVP, in which path capacities are assigned equally to path flows, and it can not provide redundant capacities to accommodate external traffic changes.

4.2.1.3 Comparison of Traffic Distribution in the LSP-CVP and the LSP-L2

Based on the optimal traffic flows and capacities assignments shown in Tables 4-4 and 4-5, the link utilization ratios for an arbitrary scenario of the LSP-CVP and the LSP-L2 (provided by the GAMS) are shown in Tables 4-6 and 4-7, respectively.

Table 4-6. Link utilization ratios of the LSP-CVP in scenario 1.

Link	Link Flow	Link Capacity	Utilization Ratio
1	97.00	100	0.97
2	97.00	100	0.97
3	93.00	100	0.93
4	0	86	0
5	100.00	100	1.00
6	94.00	100	0.94
7	22.00	100	0.22
8	90.00	100	0.90
9	80.00	100	0.80
10	100.00	100	1.00
11	54.00	100	0.54
12	11.00	60	0.18
13	35.00	100	0.35
14	33.00	33	1.00
15	99.00	100	0.99
16	68.00	68	1.00
17	100.00	100	1.00

Table 4-7. Link utilization ratios of the LSP-L2 in scenario 1.

Link	Link Flow	Link Capacity	Utilization Ratio
1	84.93	100	0.849
2	84.93	100	0.849
3	84.93	100	0.849
4	18.92	86	0.220
5	84.93	100	0.849
6	74.11	100	0.741
7	34.08	100	0.341
8	71.07	100	0.711
9	78.62	100	0.786
10	84.93	100	0.849
11	61.10	100	0.611
12	26.07	60	0.435
13	84.93	100	0.849
14	28.03	33	0.849
15	57.75	100	0.578
16	57.75	68	0.849
17	84.93	100	0.849

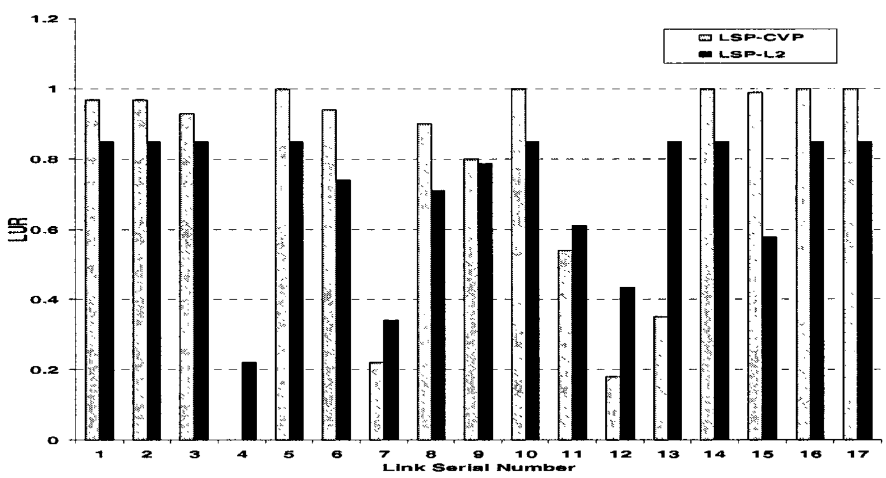


Figure 4-4. Comparison of link utilization ratios (LURs) of the LSP-CVP and the LSP-L2 in scenario 1.

Fig. 4-4 compares the LURs of a scenario for each link of the LSP-CVP and the LSP-L2 based on the Tables 4-6 and 4-7. As mentioned before, we choose the SDLUR to express the spreading of traffic distribution. The lower standard deviation corresponds to the more even traffic distribution. For the LSP-CVP, in Fig. 4-4, the mean value of LURs in scenario 1 is 0.753 and the SDLUR is 0.337. Correspondingly, for the LSP-L2, the mean value of LURs is 0.710 and the SDLUR is 0.197. As the SDLUR of the LSP-L2 is lower than that of the LSP-CVP, the LURs are spread out more evenly for the LSP-L2.

Fig. 4-5 shows the optimized k_p for each path in an arbitrary scenario. It is observed that the optimum k_p's are between 6 and 20. As shown in Fig. 4-2, for these k_p's, most traffic flows are transmitted via the L2-CTS LSPs. In other words, we can only consider the L2-CTS transmission for each OD pair to implement the optimization and ignore the Layer 3 default routing transmission. Therefore, it balances the resource utilization and decreases the possibility of congestion in the network.

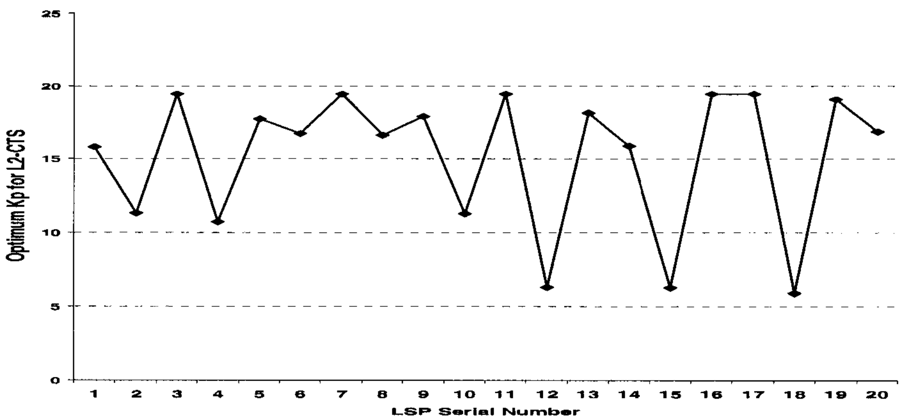


Figure 4-5. Optimized k_p for each L2-CTS LSP (available network resources that can be assigned on L2-CTS transmission) in scenario 1.

The results shown in Tables 4-6, 4-7 and Figs. 4-4, 4-5 have been obtained for one scenario only. To evaluate the performance more thoroughly, we have chosen 20 simulation scenarios for each case. For each OD pair w, we have used a Poisson random number generator to generate 20 sets of numbers with mean value that equals to the corresponding external traffic flow number that was generated in Step 2. Each set of Poission random numbers consists of one simulation scenario. Because we have chosen

10 OD pairs in the prototype network-1, there are 10 groups of Poission random numbers (each group includes 20 sets) as shown in Table 4-8.

Table 4-8. Arbitrarily 20 sets of external traffic demands. N is the scenario number, and γ_w is the external traffic demand.

N	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8	γ_9	γ_{10}
1	36	61	46	65	58	78	54	53	36	70
2	32	58	44	61	53	79	61	42	39	63
3	20	56	36	68	47	83	28	41	26	75
4	30	56	34	66	60	82	52	55	37	73
5	39	37	31	62	48	75	55	44	33	76
6	29	50	55	46	62	71	44	52	34	71
7	33	53	51	61	47	77	48	52	42	59
8	22	63	38	84	58	67	49	46	36	55
9	29	49	37	59	50	67	47	41	30	73
10	48	52	37	55	54	78	57	65	43	90
11	38	49	48	48	55	66	52	52	42	60
12	23	59	36	50	61	68	52	58	32	62
13	46	76	42	53	46	62	48	53	39	46
14	28	63	40	66	66	71	57	50	36	59
15	24	48	51	60	45	84	42	47	43	53
16	37	61	44	54	70	67	51	56	21	71
17	34	48	40	66	50	75	46	45	47	67
18	31	40	32	59	55	69	51	53	28	65
19	32	46	38	78	59	73	50	46	39	63
20	29	55	53	68	44	69	52	50	46	56

The comparison of SDLURs for each scenario of one case is shown in Fig. 4-6. It is observed that the SDLURs of the LSP-L2 are smaller than those of the LSP-CVP in each scenario. In other words, if the traffic intensity $\rho \leq 1$, the cut-through switching rate R_{L2-CTS} for L2-CTS approaches one when k_p is large enough, and hence we can ignore the traffic flow that is transmitted by the Layer 3 default routing. Thus, the traffic flow distribution on each link of the LSP-L2 is more even than the LSP-CVP. In the network, the higher the maximum load on any specific link, the more catastrophic will be the effect of the link failure. Therefore, if the traffic flow can be distributed evenly and the highest link load can be reduced, the network can have high robustness.

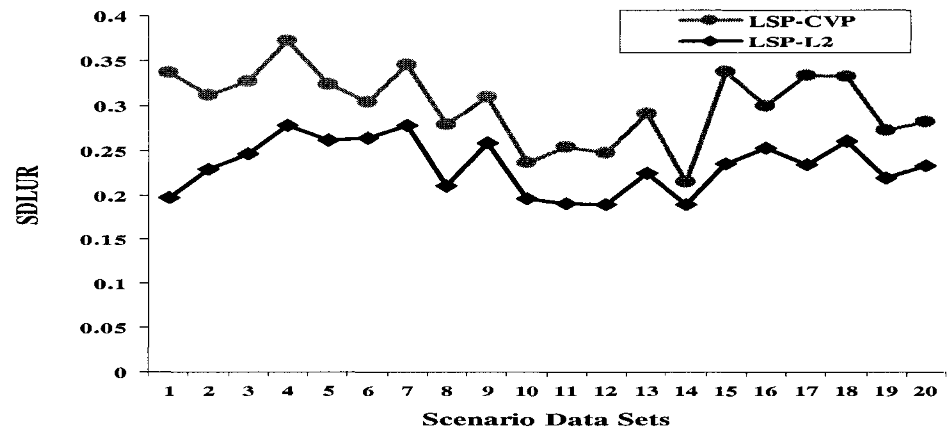


Figure 4-6. Standard deviation of link utilization ratio (SDLUR) of the LSP-CVP and the LSP-L2 of one case (20 scenarios).

We have simulated 20 different cases, each with 20 scenarios. The comparison of SDLURs is shown in Fig. 4-7. The SDLUR in each case is calculated by averaging values from 20 different sets of experimental scenarios. It can be calculated from Fig. 4-7 that the overall average SDLUR (20 cases, 400 total scenarios) of the LSP-L2 is 18.11% less than that of the LSP-CVP. It means that the LSP-L2 distributes the traffic flow more evenly than the LSP-CVP. Furthermore, we can conclude that the LSP-L2 has higher robustness than the LSP-CVP, because a network normally can have high robustness if the traffic flow can be distributed evenly and the highest link load can be reduced.

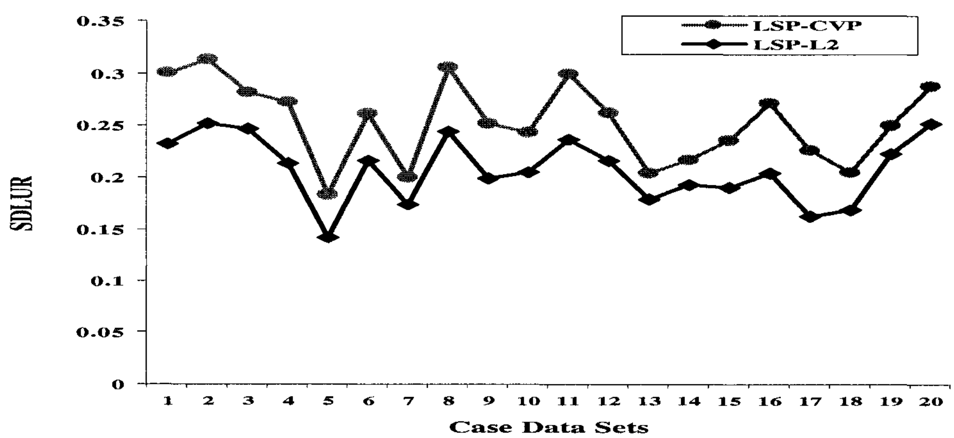


Figure 4-7. SDLUR of the LSP-CVP and LSP-L2 of 20 cases.

4.2.2 Performance Evaluation in Prototype Network-2

In order to investigate the proposed L2-CTS LSP model and the incorporated LSP flow distribution objective function on a large-scale problem, we now apply both the LSP-CVP and the LSP-L2 on a more complex prototype network-2, which has 20 nodes and 35 links as shown in Fig. 4-8, to simulate the distribution of traffic flows and assign the optimal capacities for different L2-CTS LSPs. In each experiment, we choose 25 OD pairs randomly and assume that there are at most two LSPs for each OD pair, and hence there are at most 50 paths for each scenario. We also use the Dijkstra's algorithm to find these two shortest disjointed paths for each OD pair. After simulation, each path is assigned a capacity. A capacity of zero corresponds to a non-existent path.

Both the LSP-L2 and the LSP-CVP are applied to this prototype network, and the optimization is performed for both models using GAMS as well.

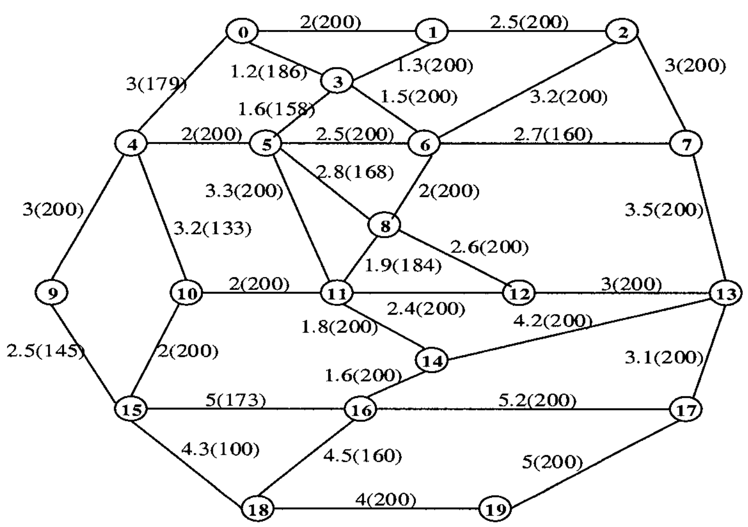


Figure 4-8. A prototype network with 20 nodes. The numbers over each link denote distance (capacity).

The main steps of the experiment are as follows:

1. Choose 25 OD pairs randomly and use Dijkstra's algorithm to find out the 50 shortest disjointed paths.

2. For the chosen 25 OD pairs, use a random number generator to generate the set of 25 external traffic demands γ_ω. The γ_ω's are created with uniform distribution with values in the range [20, 80] requests/ time-unit.

3. For each OD pair, assume $\rho = 0.8$, $N = 20$, and $\beta = 0.001$

4. Use GAMS to perform the optimization.

5. For each OD pair w, the arrival traffic flow follows a Poisson distribution. Use a Poisson random number generator to generate 30 sets of numbers with mean value that equals to the corresponding external traffic flow number that was generated in Step 2. Each set of Poission random numbers consists of one simulation scenario.

6. Repeat Step 4.

Table 4-9. Network parameters used in the simulation. C_i is the capacity of link i, and γ_w is the input flow for OD pair w.

C_i, $i = 1,\ldots,35$					γ_w, $w = 1,\ldots,25$			
C_1	200	C_{19}	184		γ_1	50	γ_{19}	49
C_2	200	C_{20}	200		γ_2	77	γ_{20}	42
C_3	179	C_{21}	200		γ_3	39	γ_{21}	23
C_4	186	C_{22}	200		γ_4	20	γ_{22}	52
C_5	200	C_{23}	200		γ_5	37	γ_{23}	67
C_6	200	C_{24}	145		γ_6	55	γ_{24}	49
C_7	200	C_{25}	200		γ_7	35	γ_{25}	27
C_8	158	C_{26}	200		γ_8	74		
C_9	200	C_{27}	200		γ_9	24		
C_{10}	200	C_{28}	200		γ_{10}	41		
C_{11}	200	C_{29}	200		γ_{11}	58		
C_{12}	160	C_{30}	173		γ_{12}	47		
C_{13}	200	C_{31}	200		γ_{13}	63		
C_{14}	133	C_{32}	100		γ_{14}	62		
C_{15}	200	C_{33}	160		γ_{15}	38		
C_{16}	168	C_{34}	200		γ_{16}	31		
C_{17}	200	C_{35}	200		γ_{17}	61		
C_{18}	200				γ_{18}	32		

Table 4-10. OD pairs, paths, routes and their costs.

OD Pair Index	OD Pair	Path Index	Routes and Nodes	Path Cost (Distance)
1	(0, 6)	P1	{0, 3, 6}	2.7
		P2	{0, 1, 3, 5, 6}	7.4
2	(0, 10)	P3	{0, 4, 10}	6.2
		P4	{0, 3, 5, 11, 10}	8.1
3	(2, 8)	P5	{2, 6, 8}	5.2
		P6	{2, 1, 3, 5, 8}	8.2
4	(3, 8)	P7	{3, 5, 8}	4.4
		P8	{3, 6, 8}	3.5
5	(4, 7)	P9	{4, 5, 6, 7}	7.2
		P10	{4, 0, 1, 2, 7}	10.5
6	(5, 15)	P11	{5, 4, 10, 15}	7.2
		P12	{5, 3, 0, 4, 9, 15}	11.3
7	(7, 10)	P13	{7, 6, 8, 11, 10}	8.6
		P14	{7, 2, 1, 3, 5, 4, 10}	13.6
8	(1, 13)	P15	{1, 2, 7, 13}	9.0
		P16	{1, 3, 6, 8, 12, 13}	10.4
9	(10, 14)	P17	{10, 11, 14}	3.8
		P18	{10, 15, 16, 14}	8.6
10	(9, 14)	P19	{9, 15, 10, 11, 14}	8.3
		P20	{9, 4, 5, 8, 12, 13, 14}	17.6
11	(11, 17)	P21	{11, 14, 16, 17}	8.6
		P22	{11, 12, 13, 17}	8.5
12	(12, 16)	P23	{12, 11, 14, 16}	5.8
		P24	{12, 13, 17, 16}	11.3
13	(9, 18)	P25	{9, 15, 18}	6.8
		P26	{9, 4, 10, 11, 14, 16, 18}	16.1
14	(13, 19)	P27	{13, 17, 19}	8.1
		P28	{13, 14, 16, 18, 19}	14.3
15	(14, 19)	P29	{14, 16, 18, 19}	10.1
		P30	{14, 13, 17, 19}	12.3
16	(8, 18)	P31	{8, 11, 14, 16, 18}	9.8
		P32	{8, 12, 11, 10, 15, 18}	13.3
17	(10, 17)	P33	{10, 15, 16, 17}	12.2
		P34	{10, 11, 12, 13, 17}	10.5
18	(1, 16)	P35	{1, 3, 5, 11, 14, 16}	9.6
		P36	{1, 0, 4, 10, 15, 16}	15.2
19	(5, 17)	P37	{5, 11, 14, 16, 17}	11.9
		P38	{5, 8, 12, 13, 17}	11.5
20	(2, 11)	P39	{2, 6, 8, 11}	7.1
		P40	{2, 1, 3, 5, 11}	8.7
21	(0, 2)	P41	{0, 1, 2}	4.5
		P42	{0, 3, 6, 2}	5.9
22	(2, 9)	P43	{2, 1, 3, 5, 4, 9}	10.4
		P44	{2, 6, 8, 11, 10, 15, 9}	13.6
23	(13, 18)	P45	{13, 14, 16, 18}	10.3
		P46	{13, 17, 19, 18}	12.1
24	(9, 17)	P47	{9, 15, 16, 17}	12.7
		P48	{9, 4, 5, 8, 12, 13, 17}	16.5
25	(3, 19)	P49	{3, 6, 7, 13, 17, 19}	15.8
		P50	{3, 5, 11, 14, 16, 18, 19}	16.8

Table 4-9 shows some network parameters that we are using in the simulation. Table 4-10 shows all paths and routes associated with OD pairs and the corresponding path costs. Tables 4-11 and 4-12 show the path and link incidence.

Table 4-11. Links and paths incidence. "*" is used to indicate the corresponding path passing the link.

Path / Link	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1		*								*															
2						*				*				*	*										
3			*							*		*													
4	*			*								*													
5		*				*								*		*									
6					*																				
7										*				*	*										
8		*		*		*	*					*		*											
9	*							*								*									
10									*		*			*						*					
11		*							*																
12									*				*												
13												*								*					
14			*								*			*											
15				*																					
16						*	*													*					
17					*			*					*			*									
18															*										
19													*												
20																*				*					
21				*									*				*		*						
22																						*	*		
23																*				*		*		*	
24												*							*						*
25											*							*	*						
26																	*		*		*		*		
27																				*					
28																						*		*	
29																		*			*		*		
30																		*							
31																					*			*	
32																									*
33																									
34																									
35																									

Table 4-12. Links and paths incidence. "*" is used to indicate the corresponding path passing the link. (Continued)

Path / Link	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1											*					*									
2															*	*		*							
3											*														
4																	*								
5										*					*			*							
6														*			*		*						
7																									
8										*					*			*							*
9																	*							*	
10																		*					*		
11																									
12																								*	
13	*						*											*					*		
14	*						*				*														
15										*		*			*										*
16													*										*		
17														*					*						
18																								*	
19						*								*					*						
20													*										*		
21	*						*		*										*						
22									*																
23									*				*										*		
24																			*			*			
25								*			*								*						
26	*					*	*			*		*													*
27			*		*															*					
28		*			*				*				*								*		*	*	
29	*		*	*		*	*			*		*								*					*
30								*			*											*			
31								*				*										*			
32																									
33	*		*	*		*	*													*					*
34		*			*																*			*	
35			*	*																	*				*

We have performed the optimization for both the LSP-CVP and the LSP-L2 using the GAMS.

4.2.2.1 Performance of the LSP-CVP

The objective function for the LSP-CVP optimization is given by Eq. (3-1). In order to perform this optimization, we change the limits of the sum in Eq. (3-1) according to the parameters of the prototype network as shown in Fig. 4-8. Thus, the reformulated objective function of the LSP-CVP is as follows:

$$\text{Minimize} \quad Y = \sum_{p=1}^{50} d_p x_p \tag{4-3}$$

Subject to the following constraints:

<u>Constraints obtained from Eq. (3-2a):</u>

$$x_1 + x_2 = \gamma_1$$
$$x_3 + x_4 = \gamma_2$$
$$x_5 + x_6 = \gamma_3$$
$$x_7 + x_8 = \gamma_4$$
$$x_9 + x_{10} = \gamma_5$$
$$x_{11} + x_{12} = \gamma_6$$
$$x_{13} + x_{14} = \gamma_7$$
$$x_{15} + x_{16} = \gamma_8$$
$$x_{17} + x_{18} = \gamma_9$$
$$x_{19} + x_{20} = \gamma_{10}$$
$$x_{21} + x_{22} = \gamma_{11}$$
$$x_{23} + x_{24} = \gamma_{12}$$
$$x_{25} + x_{26} = \gamma_{13}$$
$$x_{27} + x_{28} = \gamma_{14}$$
$$x_{29} + x_{30} = \gamma_{15}$$
$$x_{31} + x_{32} = \gamma_{16}$$
$$x_{33} + x_{34} = \gamma_{17}$$
$$x_{35} + x_{36} = \gamma_{18}$$
$$x_{37} + x_{38} = \gamma_{19}$$
$$x_{39} + x_{40} = \gamma_{20}$$
$$x_{41} + x_{42} = \gamma_{21}$$
$$x_{43} + x_{44} = \gamma_{22}$$
$$x_{45} + x_{46} = \gamma_{23}$$
$$x_{47} + x_{48} = \gamma_{24}$$
$$x_{49} + x_{50} = \gamma_{25}$$

<u>Constraints obtained from Eq. (3-2b):</u>

$$x_2 + x_{10} + x_{36} + x_{41} \le C_1$$
$$x_6 + x_{10} + x_{14} + x_{15} + x_{40} + x_{41} + x_{43} \le C_2$$

$$x_3 + x_{10} + x_{12} + x_{36} \leq C_3$$
$$x_1 + x_4 + x_{12} + x_{42} \leq C_4$$
$$x_2 + x_6 + x_{14} + x_{16} + x_{35} + x_{40} + x_{43} \leq C_5$$
$$x_5 + x_{39} + x_{42} + x_{44} \leq C_6$$
$$x_{10} + x_{14} + x_{15} \leq C_7$$
$$x_2 + x_4 + x_6 + x_7 + x_{12} + x_{14} + x_{35} + x_{40} + x_{43} + x_{50} \leq C_8$$
$$x_1 + x_8 + x_{16} + x_{42} + x_{49} \leq C_9$$
$$x_9 + x_{11} + x_{14} + x_{20} + x_{43} + x_{48} \leq C_{10}$$
$$x_2 + x_9 \leq C_{11}$$
$$x_9 + x_{13} + x_{49} \leq C_{12}$$
$$x_{12} + x_{20} + x_{26} + x_{32} + x_{43} + x_{48} \leq C_{13}$$
$$x_3 + x_{11} + x_{14} + x_{26} + x_{32} + x_{36} \leq C_{14}$$
$$x_4 + x_{35} + x_{37} + x_{40} + x_{50} \leq C_{15}$$
$$x_6 + x_7 + x_{20} + x_{38} + x_{48} \leq C_{16}$$
$$x_5 + x_8 + x_{13} + x_{16} + x_{39} + x_{44} \leq C_{17}$$
$$x_{15} + x_{49} \leq C_{18}$$
$$x_{13} + x_{31} + x_{39} + x_{44} \leq C_{19}$$
$$x_{16} + x_{20} + x_{38} + x_{48} \leq C_{20}$$
$$x_4 + x_{13} + x_{17} + x_{19} + x_{26} + x_{32} + x_{34} + x_{44} \leq C_{21}$$
$$x_{22} + x_{23} + x_{34} \leq C_{22}$$
$$x_{16} + x_{20} + x_{22} + x_{24} + x_{34} + x_{38} + x_{48} \leq C_{23}$$
$$x_{12} + x_{19} + x_{25} + x_{44} + x_{47} \leq C_{24}$$
$$x_{11} + x_{18} + x_{19} + x_{33} + x_{36} + x_{44} \leq C_{25}$$
$$x_{17} + x_{19} + x_{21} + x_{23} + x_{26} + x_{31} + x_{32} + x_{35} + x_{37} + x_{50} \leq C_{26}$$
$$x_{20} + x_{28} + x_{30} + x_{45} \leq C_{27}$$
$$x_{22} + x_{24} + x_{27} + x_{30} + x_{34} + x_{38} + x_{46} + x_{48} + x_{49} \leq C_{28}$$
$$x_{18} + x_{21} + x_{23} + x_{26} + x_{28} + x_{29} + x_{31} + x_{32} + x_{35} + x_{37} + x_{45} + x_{50} \leq C_{29}$$
$$x_{18} + x_{33} + x_{36} + x_{47} \leq C_{30}$$
$$x_{21} + x_{24} + x_{33} + x_{37} + x_{47} \leq C_{31}$$
$$x_{25} \leq C_{32}$$
$$x_{26} + x_{28} + x_{29} + x_{31} + x_{32} + x_{45} + x_{50} \leq C_{33}$$
$$x_{27} + x_{30} + x_{46} + x_{49} \leq C_{34}$$
$$x_{28} + x_{29} + x_{46} + x_{50} \leq C_{35}$$

and $x_p \geq 0$, $p = 1,...,50$.

After performing the optimization using GAMS, the result for an arbitrary scenario is shown in Table 4-13. It shows the assigned traffic flows for each path in the LSP-CVP. The minimized overall transmission delay $Y = 10010.85$ (as calculated by Eq. (4-3), with d_p as shown in Table 4-10).

Table 4-13. Optimized path flows for the LSP-CVP in scenario 1.

x_1	55.00	x_{11}	60.00	x_{21}	0	x_{31}	35.00	x_{41}	30.00
x_2	0	x_{12}	0	x_{22}	54.00	x_{32}	0	x_{42}	0
x_3	73.00	x_{13}	33.00	x_{23}	40.00	x_{33}	51.00	x_{43}	51.00
x_4	3.00	x_{14}	0	x_{24}	0	x_{34}	3.00	x_{44}	0
x_5	37.00	x_{15}	69.00	x_{25}	60.00	x_{35}	29.00	x_{45}	54.00
x_6	0	x_{16}	0	x_{26}	0	x_{36}	0	x_{46}	6.00
x_7	0	x_{17}	29.00	x_{27}	63.00	x_{37}	0	x_{47}	39.00
x_8	17.00	x_{18}	0	x_{28}	0	x_{38}	51.00	x_{48}	2.00
x_9	48.00	x_{19}	46.00	x_{29}	42.00	x_{39}	42.00	x_{49}	21.00
x_{10}	0	x_{20}	0	x_{30}	0	x_{40}	0	x_{50}	0

As we mentioned before, when the CVP model is used for LSP optimization, the optimized traffic flows are considered as the optimal LSP capacity assignment. For instance, it is observed that the flow in Path 7 is zero, which indicates that Path 7 does not exist and all the traffic for OD pair (3,8) is carried by Path 8. Because path capacities and path flows are considered to be the same, there is no redundant capacity assigned to accommodate the external traffic changes. Consequently, the resulting LSP system is not resilient to traffic changes. Moreover, the traffic is not distributed evenly in the LSP-CVP as the nature of the shortest path algorithm.

4.2.2.2 Performance of the LSP-L2

The objective function for the LSP-L2 optimization is given by Eq. (3-7). In order to perform this optimization, we change the limits of the sum in Eq. (3-7) according to the parameters of the prototype network as shown in Fig. 4-8. Thus, the reformulated objective function of the LSP-L2 is as follows:

Minimize $$Z = \sum_{p=1}^{50} d_p \frac{1-\rho^{k_p}}{1-\rho^{k_p+1}}(x_p) \qquad (4\text{-}4)$$

Subject to the following constraints:

Constraints obtained from Eq. (3-8a):

$$x_1 + x_2 = \gamma_1$$
$$x_3 + x_4 = \gamma_2$$
$$x_5 + x_6 = \gamma_3$$
$$x_7 + x_8 = \gamma_4$$
$$x_9 + x_{10} = \gamma_5$$
$$x_{11} + x_{12} = \gamma_6$$
$$x_{13} + x_{14} = \gamma_7$$
$$x_{15} + x_{16} = \gamma_8$$
$$x_{17} + x_{18} = \gamma_9$$
$$x_{19} + x_{20} = \gamma_{10}$$
$$x_{21} + x_{22} = \gamma_{11}$$
$$x_{23} + x_{24} = \gamma_{12}$$
$$x_{25} + x_{26} = \gamma_{13}$$
$$x_{27} + x_{28} = \gamma_{14}$$
$$x_{29} + x_{30} = \gamma_{15}$$
$$x_{31} + x_{32} = \gamma_{16}$$
$$x_{33} + x_{34} = \gamma_{17}$$
$$x_{35} + x_{36} = \gamma_{18}$$
$$x_{37} + x_{38} = \gamma_{19}$$
$$x_{39} + x_{40} = \gamma_{20}$$
$$x_{41} + x_{42} = \gamma_{21}$$
$$x_{43} + x_{44} = \gamma_{22}$$
$$x_{45} + x_{46} = \gamma_{23}$$
$$x_{47} + x_{48} = \gamma_{24}$$
$$x_{49} + x_{50} = \gamma_{25}$$

Constraints obtained from Eq. (3-8b):

$$N_2 + N_{10} + N_{36} + N_{41} \le C_1$$
$$N_6 + N_{10} + N_{14} + N_{15} + N_{40} + N_{41} + N_{43} \le C_2$$
$$N_3 + N_{10} + N_{12} + N_{36} \le C_3$$
$$N_1 + N_4 + N_{12} + N_{42} \le C_4$$

$$N_2 + N_6 + N_{14} + N_{16} + N_{35} + N_{40} + N_{43} \le C_5$$
$$N_5 + N_{39} + N_{42} + N_{44} \le C_6$$
$$N_{10} + N_{14} + N_{15} \le C_7$$
$$N_2 + N_4 + N_6 + N_7 + N_{12} + N_{14} + N_{35} + N_{40} + N_{43} + N_{50} \le C_8$$
$$N_1 + N_8 + N_{16} + N_{42} + N_{49} \le C_9$$
$$N_9 + N_{11} + N_{14} + N_{20} + N_{43} + N_{48} \le C_{10}$$
$$N_2 + N_9 \le C_{11}$$
$$N_9 + N_{13} + N_{49} \le C_{12}$$
$$N_{12} + N_{20} + N_{26} + N_{32} + N_{43} + N_{48} \le C_{13}$$
$$N_3 + N_{11} + N_{14} + N_{26} + N_{32} + N_{36} \le C_{14}$$
$$N_4 + N_{35} + N_{37} + N_{40} + N_{50} \le C_{15}$$
$$N_6 + N_7 + N_{20} + N_{38} + N_{48} \le C_{16}$$
$$N_5 + N_8 + N_{13} + N_{16} + N_{39} + N_{44} \le C_{17}$$
$$N_{15} + N_{49} \le C_{18}$$
$$N_{13} + N_{31} + N_{39} + N_{44} \le C_{19}$$
$$N_{16} + N_{20} + N_{38} + N_{48} \le C_{20}$$
$$N_4 + N_{13} + N_{17} + N_{19} + N_{26} + N_{32} + N_{34} + N_{44} \le C_{21}$$
$$N_{22} + N_{23} + N_{34} \le C_{22}$$
$$N_{16} + N_{20} + N_{22} + N_{24} + N_{34} + N_{38} + N_{48} \le C_{23}$$
$$N_{12} + N_{19} + N_{25} + N_{44} + N_{47} \le C_{24}$$
$$N_{11} + N_{18} + N_{19} + N_{33} + N_{36} + N_{44} \le C_{25}$$
$$N_{17} + N_{19} + N_{21} + N_{23} + N_{26} + N_{31} + N_{32} + N_{35} + N_{37} + N_{50} \le C_{26}$$
$$N_{20} + N_{28} + N_{30} + N_{45} \le C_{27}$$
$$N_{22} + N_{24} + N_{27} + N_{30} + N_{34} + N_{38} + N_{46} + N_{48} + N_{49} \le C_{28}$$
$$N_{18} + N_{21} + N_{23} + N_{26} + N_{28} + N_{29} + N_{31} + N_{32} + N_{35} + N_{37} + N_{45} + N_{50} \le C_{29}$$
$$N_{18} + N_{33} + N_{36} + N_{47} \le C_{30}$$
$$N_{21} + N_{24} + N_{33} + N_{37} + N_{47} \le C_{31}$$
$$N_{25} \le C_{32}$$
$$N_{26} + N_{28} + N_{29} + N_{31} + N_{32} + N_{45} + N_{50} \le C_{33}$$
$$N_{27} + N_{30} + N_{46} + N_{49} \le C_{34}$$
$$N_{28} + N_{29} + N_{46} + N_{50} \le C_{35}$$

Constraints obtained from Eq. (3-8c):

$$\frac{x_p}{N_p} \le \frac{2k_p}{2k_p - \ln(\beta)}, \; p = 1,\ldots,50.$$

and $x_p \geq 0$, $N_p \geq 0$, $k_p \geq 0$, $p = 1,...,50$.

After performing the optimization using GAMS, the result for an arbitrary scenario is shown in Table 4-14. It shows the assigned flow and capacity for each path in the LSP-L2.

Table 4-14. Optimized path flow and capacity assignments for the LSP-L2 in scenario 1.

Path Flow		Path Capacity		Path Flow		Path Capacity	
x_1	54.42	N_1	82.76	x_{26}	0	N_{26}	0
x_2	0.58	N_2	0.88	x_{27}	47.21	N_{27}	55.66
x_3	34.31	N_3	40.45	x_{28}	15.79	N_{28}	18.62
x_4	41.69	N_4	63.41	x_{29}	42.00	N_{29}	49.52
x_5	37.00	N_5	56.27	x_{30}	0	N_{30}	0
x_6	0	N_6	0	x_{31}	35.00	N_{31}	41.27
x_7	5.04	N_7	7.66	x_{32}	0	N_{32}	0
x_8	11.96	N_8	18.19	x_{33}	54.00	N_{33}	63.67
x_9	31.54	N_9	37.19	x_{34}	0	N_{34}	0
x_{10}	16.46	N_{10}	25.03	x_{35}	0	N_{35}	0
x_{11}	38.01	N_{11}	48.45	x_{36}	29.00	N_{36}	44.10
x_{12}	21.99	N_{12}	25.92	x_{37}	45.04	N_{37}	53.09
x_{13}	33.00	N_{13}	50.19	x_{38}	5.96	N_{38}	7.03
x_{14}	0	N_{14}	0	x_{39}	42.00	N_{39}	63.87
x_{15}	67.41	N_{15}	79.47	x_{40}	0	N_{40}	0
x_{16}	1.59	N_{16}	2.43	x_{41}	30.00	N_{41}	35.37
x_{17}	29.00	N_{17}	44.10	x_{42}	0	N_{42}	0
x_{18}	0	N_{18}	0	x_{43}	51.00	N_{43}	60.13
x_{19}	0	N_{19}	0	x_{44}	0	N_{44}	0
x_{20}	46.00	N_{20}	54.23	x_{45}	21.40	N_{45}	25.24
x_{21}	0	N_{21}	0	x_{46}	38.60	N_{46}	48.75
x_{22}	54.00	N_{22}	63.67	x_{47}	41.00	N_{47}	48.34
x_{23}	10.40	N_{23}	12.26	x_{48}	0	N_{48}	0
x_{24}	29.60	N_{24}	34.89	x_{49}	21.00	N_{49}	24.63
x_{25}	60.00	N_{25}	70.74	x_{50}	0	N_{50}	0

The minimized overall transmission delay of the LSP-L2, $Z = 9399.30$ (as calculated by Eq. (4-4), with d_p as shown in Table 4-10), whereas the minimized overall transmission delay of the LSP-CVP is 10010.85.

It is observed that the path capacity assigned for each path is always higher than the path flow (except for those path flows are zero). It means in the LSP-L2, each path is assigned some redundant capacities and the external traffic demands are satisfied by the path flows assignments. When the external traffic goes up, the LSP-L2 can accommodate some additional traffic. Hence, the LSP-L2 is resilient to traffic changes. It contrasts with the LSP-CVP, in which path capacities are assigned equally to path flows.

4.2.2.3 Comparison of Traffic Distribution in the LSP-CVP and the LSP-L2

Table 4-15. Link utilization ratios of the LSP-CVP in scenario 1.

Link	Link Flow	Link Capacity	Utilization Ratio
1	30.00	200	0.150
2	150.00	200	0.750
3	73.00	179	0.408
4	58.00	186	0.312
5	80.00	200	0.400
6	79.00	200	0.395
7	69.00	200	0.345
8	83.00	158	0.525
9	93.00	200	0.465
10	161.00	200	0.805
11	48.00	200	0.240
12	102.00	160	0.637
13	53.00	200	0.265
14	133.00	133	1.000
15	32.00	200	0.160
16	53.00	168	0.315
17	129.00	200	0.645
18	90.00	200	0.450
19	110.00	184	0.598
20	53.00	200	0.265
21	114.00	200	0.570
22	97.00	200	0.485
23	110.00	200	0.550
24	145.00	145	1.000
25	157.00	200	0.785
26	179.00	200	0.895
27	54.00	200	0.270
28	200.00	200	1.000
29	200.00	200	1.000
30	90.00	173	0.520
31	90.00	200	0.450
32	60.00	200	0.300
33	131.00	160	0.819
34	90.00	200	0.450
35	48.00	200	0.240

Based on the optimal traffic flows and capacities assignments as shown in Tables 4-13 and 4-14, the LURs for an arbitrary scenario of the LSP-CVP and the LSP-L2 are shown in Tables 4-15 and 4-16, respectively.

Table 4-16. Link utilization ratios of the LSP-L2 in scenario 1.

Link	Link Flow	Link Capacity	Utilization Ratio
1	76.040	200	0.380
2	164.864	200	0.824
3	101.750	179	0.568
4	118.098	186	0.635
5	53.176	200	0.266
6	79.000	200	0.395
7	83.864	200	0.419
8	120.296	158	0.761
9	67.978	200	0.340
10	166.556	200	0.833
11	32.123	200	0.161
12	64.541	160	0.403
13	118.985	200	0.595
14	101.321	133	0.762
15	86.730	200	0.434
16	56.999	168	0.339
17	125.560	200	0.628
18	67.405	200	0.337
19	110.000	184	0.598
20	53.559	200	0.268
21	103.694	200	0.518
22	64.401	200	0.322
23	137.157	200	0.686
24	122.985	145	0.848
25	121.015	200	0.605
26	119.438	200	0.597
27	83.197	200	0.416
28	169.635	200	0.848
29	169.635	200	0.848
30	124.000	173	0.717
31	169.635	200	0.848
32	60.000	200	0.300
33	114.197	160	0.714
34	80.073	200	0.400
35	90.660	200	0.453

Fig. 4-9 compares the LURs of a scenario for each link of the LSP-CVP and the LSP-L2 based on the Tables 4-15 and 4-16. The same as before, we choose the SDLUR to express the spreading of traffic distribution. The lower the standard deviation, the higher the evenness of the traffic distribution we can achieve. In Fig. 4-9, for the LSP-CVP, the mean value of LURs in scenario 1 is 0.528 and the SDLUR is 0.253. Correspondingly, for the LSP-L2, the mean value of LURs is 0.545 and the SDLUR is 0.201. As the SDLUR of the LSP-L2 is lower than that of the LSP-CVP, the LURs are spread out more evenly for the LSP-L2.

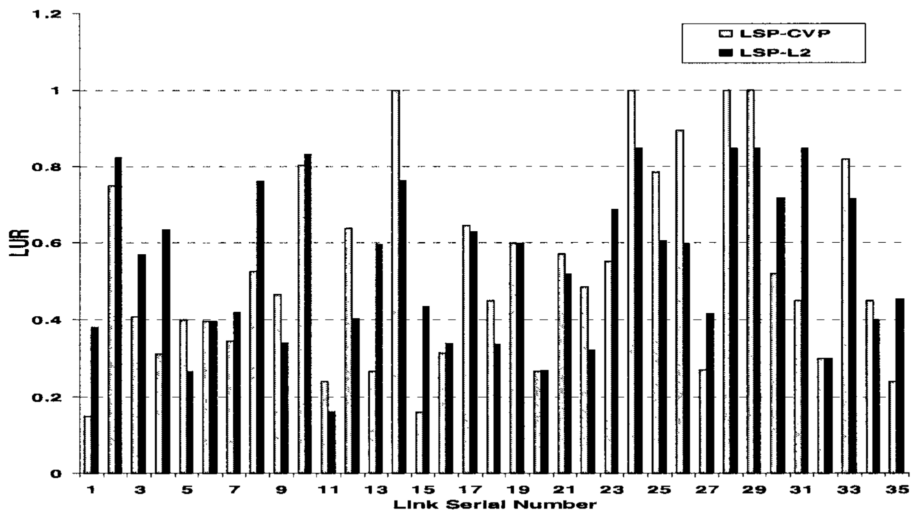


Figure 4-9. Comparison of link utilization ratios (LURs) of the LSP-CVP and the LSP-L2 in scenario 1.

Fig. 4-10 shows the optimized k_p for each path in an arbitrary scenario. It is observed that the optimum k_p's are between 6 and 20. As shown in Fig. 4-2, for these k_p's, most traffic flows are transmitted via the L2-CTS LSPs. In other words, we can only consider the L2-CTS transmission for each OD pair to implement the optimization and ignore the Layer 3 default routing transmission. Hence, it balances the resource utilization and decreases the possibility of congestion in the network.

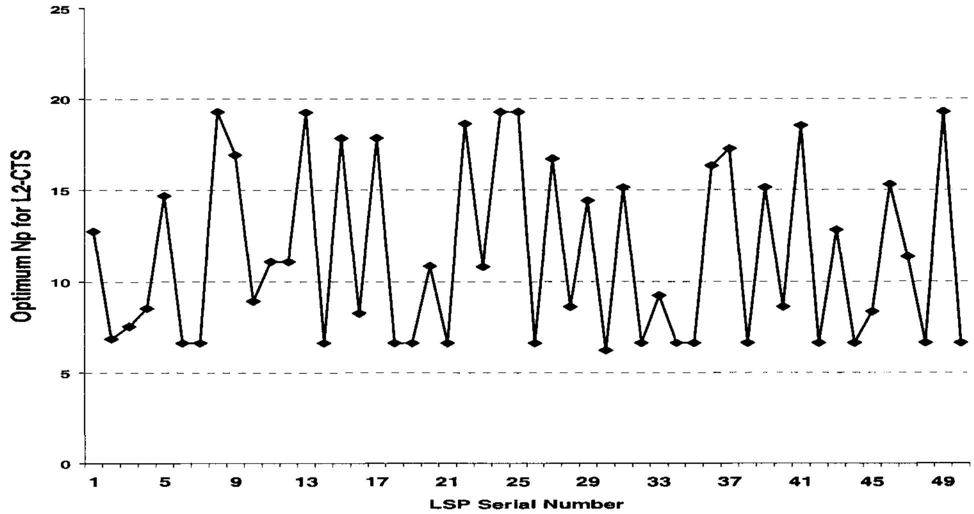


Figure 4-10. Optimized k_p for each L2-CTS LSP (available network resources that can be assigned on L2-CTS transmission) in scenario 1.

Table 4-17. Arbitrarily 30 sets of external traffic demands. N is the scenario number, and γ_w is the external traffic demand.

N \ γ_w	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	55	76	37	17	48	60	33	69	29	46	54	40	60	63	42	35	54	29	51	42	30	51	60	41	21
2	61	84	42	20	54	66	37	76	33	51	60	45	66	69	47	39	59	33	56	46	34	58	65	45	24
3	48	83	39	11	57	60	28	66	29	39	60	28	54	64	39	43	51	35	48	41	22	48	58	36	20
4	55	81	50	26	48	42	35	73	32	52	49	35	52	58	47	37	50	21	41	51	28	53	57	40	18
5	48	79	33	11	53	51	31	65	33	51	47	47	53	54	45	30	53	27	47	48	34	54	56	45	17
6	51	80	40	19	58	56	18	60	32	48	48	42	50	55	35	29	46	34	60	38	22	53	66	42	19
7	54	84	44	17	57	74	29	67	25	42	46	46	60	46	42	33	42	27	49	42	29	49	65	39	28
8	49	63	39	6	56	58	41	70	23	52	55	33	61	65	44	40	68	30	59	37	23	48	69	40	25
9	53	67	31	17	47	67	33	70	25	38	61	49	68	63	52	35	53	25	52	46	29	37	66	47	19
10	69	62	28	7	53	61	39	84	26	45	55	38	66	61	44	29	52	17	47	36	32	54	49	35	15
11	60	77	38	21	41	66	22	74	34	44	47	46	69	65	41	31	52	30	64	34	36	60	64	53	28
12	61	74	37	13	51	69	37	67	20	33	41	35	66	64	46	28	63	17	43	44	33	44	61	44	24
13	60	69	38	15	44	67	28	75	41	37	47	37	59	56	52	31	55	30	44	33	35	55	57	46	17
14	43	73	40	16	42	62	43	74	21	45	41	49	66	68	37	35	54	16	56	47	29	45	53	26	19
15	62	95	36	18	49	41	21	57	29	40	63	37	51	49	39	27	49	28	47	45	29	49	60	37	24
16	57	81	40	9	46	75	42	68	30	48	48	35	58	68	37	35	54	32	52	48	31	54	50	45	31
17	51	80	35	12	50	60	33	79	34	36	53	33	68	55	47	37	47	30	43	38	34	50	62	42	20
18	55	81	34	13	50	68	29	61	31	51	42	39	58	61	43	41	55	28	52	44	31	45	49	46	17
19	76	72	38	15	43	61	33	84	30	37	48	42	72	60	42	37	44	27	67	43	34	55	63	45	17
20	65	72	30	16	46	52	34	73	30	42	63	37	65	74	45	46	53	31	48	48	28	56	68	44	20
21	60	84	42	15	56	71	33	70	28	47	53	31	52	62	42	33	37	24	59	51	19	54	59	38	18
22	75	91	32	22	52	62	43	61	33	58	47	57	65	64	49	41	56	26	67	46	27	49	57	44	20
23	50	79	33	17	43	68	32	65	31	49	48	42	50	74	30	46	54	29	52	40	26	44	77	42	25
24	53	83	41	23	61	57	54	62	26	56	51	37	61	66	54	29	53	39	54	43	31	47	73	38	20
25	51	72	38	18	56	58	37	60	25	39	57	37	78	62	47	33	60	36	58	30	34	50	56	37	22
26	44	70	44	17	43	72	24	72	42	44	61	32	53	55	49	44	73	20	54	44	26	47	60	38	23
27	53	79	33	20	43	55	36	71	21	29	49	42	53	42	42	29	61	27	65	43	42	40	71	42	13
28	45	73	27	17	47	62	45	64	33	40	45	40	62	70	51	33	48	29	54	36	26	48	52	43	28
29	61	72	35	16	44	58	27	71	29	43	56	44	66	57	40	34	44	27	41	38	31	60	54	41	21
30	70	78	27	17	60	67	30	73	31	48	48	35	61	44	36	23	59	30	58	39	33	56	45	28	29

The results shown in Tables 4-15, 4-16 and Figs. 4-9, 4-10 have been obtained for one scenario only. To evaluate the performance more thoroughly, we have chosen 30 simulation scenarios for each case. For each OD pair w, we have used a Poisson random number generator to generate 30 sets of numbers with mean value that equals to the corresponding external traffic flow number that was generated in Step 2. Each set of Poission random numbers consists of one simulation scenario. Those 25 groups of Poission random numbers (each group includes 30 sets) are as shown in Table 4-17.

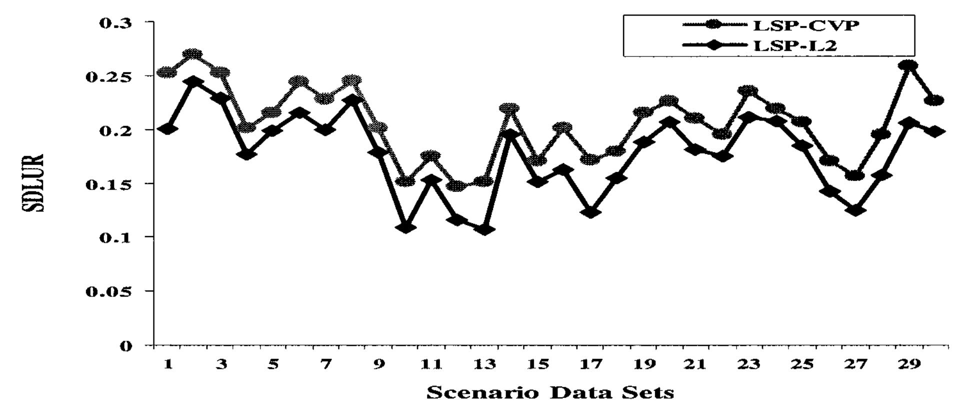


Figure 4-11. Standard deviation of link utilization ratio (SDLUR) of the LSP-CVP and the LSP-L2 of one case (30 scenarios).

The comparison of SDLURs for each scenario of one case is shown in Fig. 4-11. It is observed that the SDLURs of the LSP-L2 are smaller than those of the LSP-CVP in each scenario. In other words, if the traffic intensity $\rho \leq 1$, the cut-through switching rate R_{L2-CTS} for L2-CTS approaches one when k_p is large enough, and hence we can ignore the traffic flow that is transmitted by the Layer 3 default routing. Thus, the traffic flow distribution on each link of the LSP-L2 is more even than the LSP-CVP. As we mentioned before, the higher the maximum load on any specific link, the more catastrophic will be the effect of the link failure. Therefore, if the traffic flow can be distributed evenly and the highest link load can be reduced, the network can have high robustness.

We have simulated 20 different cases, each with 30 scenarios. The comparison of SDLURs is shown in Fig. 4-12. The SDLUR in each case is calculated by averaging values from 30 different sets of experimental scenarios. It can be calculated from Fig. 4-12 that the overall average SDLUR (20 cases, 600 total scenarios) of the LSP-L2 is 16.57% less than that of the LSP-CVP. This means the LSP-L2 distributes the traffic flow more evenly than the LSP-CVP. Furthermore, we can conclude that the LSP-L2 has higher robustness than the LSP-CVP, because a network normally can have high robustness if the traffic flow can be distributed evenly and the highest link load can be reduced.

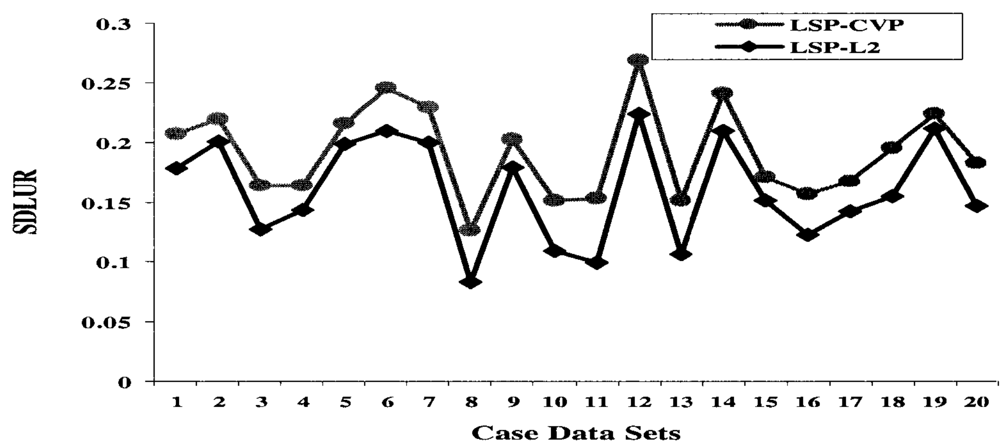


Figure 4-12. SDLUR of the LSP-CVP and LSP-L2 of 20 cases.

4.3 Characterization of Numerical Results

It has been proved [10] that the optimal routing directs traffic exclusively along *minimum first derivative length* (MFDL) paths for each OD pair. This is a typical behavior of optimal routing when Eq. (3-1) is based on the M/M/1 approximation. Hence, if there are some LSPs which have equal length to connect each OD pair, the OD pair tends to use only one path (the high-capacity path) for routing when the external traffic is low. With the increase of the traffic, the other low-capacity paths are used to avoid overloading the first one. From both Tables 4-4, 4-5 and Tables 4-13, 4-14, we observe that the LSP-L2 model has better performance compared to the LSP-CVP model under the same circumstances, and hence prevents the situation that faster paths (in terms of transmission delay) may be over-utilized.

4.4 Computation Complexity Analysis

Constraint-based Routing (CR) algorithms are normally categorized as either online or offline. Offline CR optimizes the network resource by using a priori knowledge of traffic demand. It requires a demand matrix as the input. It is very difficult, if not impossible, to obtain an accurate demand matrix in the case of dynamic MPLS LSPs setup. Hence, online CR is the appropriate approach to obtain the optimal LSP setup in MPLS networks. It can setup paths and optimize network resources in real time at the ingress node. However, for real-time implementation, the computation time should be as short as possible. Although our proposed objective function takes into account the delay parameter only, the computation complexity is still an important issue in our proposed technique.

The commonly used path constraints in CR are bandwidth, hop count, delay, jitter, loss ratio, and monetary cost. If multiple constraints are to be optimized simultaneously, the complexity of routing algorithm usually becomes very high. It has been proved that finding an optimal path subject to constraints based on two or more additive (e.g., delay, and jitter) and/or multiplicative constraints (e.g., loss ratio) in any possible combination is NP-complete [54]. However, the NP-completeness is based on the assumptions that: 1) all the constraints are independent; and 2) the delay and jitter of a link are known a priori. Because bandwidth, delay and jitter are not independent in packet networks, polynomial algorithms for finding paths with hop count, delay, jitter and buffer space constraints exist. Hence, we can convert additive and multiplicative constraints into a bandwidth combination requirement.

Consider a network with N nodes and M links. E is the number of edge nodes ($E \leq N$). If every edge node acts as both source and destination node, we have a maximum of E^2 pairs of source-destination nodes. We know that the complexity of computing the shortest disjointed paths for each source-destination pair by using Dijkstra's algorithm is $O(NM)$. Because the number of disjointed paths for a single source-destination pair can be as many as $M/2$ (i.e., $O(M)$), the worst case scenario for finding out all shortest disjointed paths for all possible source-destination pairs is $O(E^2NM^2)$. However, using a priority queue with Fibonacci heap in the implementation [55], we can improve the complexity of Dijkstra's algorithm to $O(M + N\log N)$. If all the network nodes are also the edge nodes (i.e., $E = N$), the overall computational complexity of finding out all shortest disjointed paths for all possible source-destination pairs becomes $O(N^2M^2 + N^3M\log N)$. Meanwhile, the computational complexity of the NLP is generally data dependent, and usually has a higher complexity than the shortest path algorithm as the NLP algorithm is generally heuristic. Thus, the overall complexity of our proposed technique depends on the combinatorial complexity of the NLP algorithm and the shortest path algorithm.

4.5 Summary

In this chapter, we first analyzed the Layer 2 cut-through switching transmission rate of the proposed L2-CTS LSP model. We then evaluated the performance of the LSP system resulting from the L2-CTS LSP model and the incorporated nonlinear LSP flow distribution objective function, and compared it with the LSP-CVP model in two different prototype networks. Simulation results showed that the proposed model and the objective function reduce the overall transmission delay and distribute traffic flows more evenly across different LSPs. Furthermore, we characterized the numerical results and analyzed the computation complexity of our proposed techniques for LSP optimization in MPLS networks.

Chapter 5

Conclusions and Future Research Directions

The objective of TE in the service providers' networks is to satisfy the customers' traffic demands and minimize congestion while simultaneously optimizing the network resource utilization. The MPLS plays a key role in IP networks by providing QoS as well as TE features. It reduces the complexity of packet forwarding in IP networks, and achieves the simplified connection-oriented forwarding characteristics of Layer 2 switching technologies while retaining the equally desirable flexibility and scalability of Layer 3 routing.

In this thesis, we have developed an analytical L2-CTS LSP model and an incorporated LSP traffic flow distribution objective function for the LSP capacity and flow assignment using TE in the MPLS network. The main contributions of this thesis are as follows [12]:

- A new analytical L2-CTS LSP model has been proposed for LSP optimization using TE in the MPLS network. We treat the TE issue by setting up different types of LSPs through the MPLS network. The proposed L2-CTS LSP model combines two different transmission mechanisms: Layer 2 cut-through switching and Layer 3 default routing. Using this model, we can achieve Layer 2 simplified connection-oriented forwarding while retaining Layer 3 routing flexibly and scalably.

- An incorporated LSP traffic flow distribution optimization objective function has been proposed to minimize the overall network transmission weighted delay and optimize the traffic flow distribution across different LSPs. It is a nonlinear objective function which is based on the load dependant parameters. The resulting LSP system is based on the path-node incidence, and hence is suitable for the dominant multi-class traffic flows in MPLS networks, where the traffic flow through each LSP may belong to a specific traffic class. The proposed nonlinear LSP optimization objective function can balance the resource utilization, distribute traffic flows more evenly and decrease the possibility of congestions.

5.1 Future Research Directions

In the MPLS network, an LSR creates or destroys a binding between a label and an FEC as a result of a particular event [1]. Such an event could be triggered either by the data packets that have to be forwarded by the LSR or by the control (routing) information that has to be processed by the LSR. It is referred to as *data-driven* label binding scheme when the creation or destruction of bindings is triggered by the data packets. On the other hand, when the creation or destruction of bindings is triggered by the control information, it is referred to as *control-driven* label binding scheme.

The choice between these two methods for establishing bindings depends on different situations in terms of performance, scalability and robustness. In this thesis, we have chosen the data-driven scheme to build our proposed model. In the future, the research work can be extended under the circumstance of the control-driven scheme, since the control-driven scheme might have a better performance if the network topology could be stable for a long period. Furthermore, the operation of the data-driven scheme in the real practice network usually includes a label-setup policy and an adjustable label-release timer. The label-setup policy is that the label-setup controller does not start to set up a label binding until the accumulated packets of the same flow in the buffer have exceeded a triggering threshold. The label-release timer is used to control the reserved L2-CTS LSP, and the reserved LSP is released only if no packet arrives during the maximum allowed inactive duration (label-release timer). In the future, the research work can include these operations.

In this thesis, our proposed optimization objective function only considered the network parameter in terms of the weighted transmission delay. In the future, more network constraints can be taken into account, such as jitter and loss ratio. The multiple constraints are to be optimized simultaneously although it would increase the computational complexity and make the optimization much more complicated.

In the simulation, different modeling assumptions such as traffic models, server behavior models and scheduling mechanisms, have an important impact on the performance measurement [51]. In this thesis, we have chosen the basic queueing structures to make the analysis simpler. In the future, the research work can include more complex models. For instance, the MMPP arrival can be used as the external traffic model. In order to meet the QoS objectives of each connection, an arbitration function can be added to implement the scheduling algorithm, such as the priority-based scheduling.

Bibliography

1. B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kufmann, May 2000.

2. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC-3031, January 2001.

3. S. Nakazawa, H. Tamura, K. Kawahara, and Y. Oie, "Performance Analysis of IP Datagram Transmission Delay in MPLS: Impact of Both Number and Bandwidth of LSPs of Layer 2," *IEICE Trans. Commun.,* Vol. E85-B, No. 1, January 2002.

4. D. Zheng, X. Liu, M. Mandal, and W. Lu, "Virtual Traffic Path Optimization in Connection-Oriented Networks with Stochastic Traffic," *Journal of Network and Systems Management*, Vol. 12, No. 2, June 2004.

5. R. Boutaba, W.Szeto, and Y. Iraqi, "DORA: Efficient Routing for MPLS Traffic Engineering," *Journal of Network and Systems Management*, Vol. 10, No. 3, September 2002.

6. D. Mitra and Q. Wang, "Stochastic Traffic Engineering, with Applications to Network Revenue Management," *Proc. IEEE INFOCOM*, April 2003.

7. K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications," *IEEE Journal on Selected Areas in Commun.: Quality of Service in the Internet*, Vol. 18, No. 12, pp. 921–940, December 2000.

8. B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," *Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communication Societies (IEEE INFOCOM)*, Tel Aviv, Israel, March 2000.

9. Z. Liu, Y. Sun, and X. Xue, "A Static Routing Algorithm Used in the Internet Traffic Engineering," *Proc. of the 7th Asian-Pacific Conference on Circuits and Systems (IEEE APCCAS)*, Tianjin, China, December 2000.

10. D. Bertsekas, and R. Gallager, *Data Networks*, 2nd edition, Prentice Hall, NJ, 1992.

11. X. Liu, "Capacity and Flow Assignment Issues in MPLS Networks," *International Journal of Network Management*, Vol. 13, pp. 173-179, 2003.

12. W. Lu, and M. Mandal, "Optimal LSP Capacity and Flow Assignment Using Traffic Engineering in MPLS Networks," accepted for publication in *the IEEE Global Telecommunications Conference* (*GLOBECOM'2004*), Dallas, USA, November 2004.

13. R. Callon, *A Framework for Multiprotocol Label Switching*, September 1999.

14. P. Brittain, and A. Farrel, *MPLS Traffic Engineering: A Choice of Signaling Protocols*, January 2000.

15. E. Crawley, R. Nair, B. Jajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet," RFC-2386, August 1998.

16. A. Tanenbaum, *Computer Networks*, third edition, Prentice Hall PTR, 1996.

17. Cisco's 12000 Series, http://www.cisco.com/warp/public/733/12000/

18. Ascend's GRF routers, http://www.ascend.com/300.html

19. Bay Networks' Accelar Routing Switches, http://business5.baynetworks.com/MainBody.asp

20. 3Com's switches, http://www.3com.com/products/switches.html

21. Juniper's M40 Series, http://www.juniper.net/products/default.htm

22. R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," Internet RFC 1633, January 1994.

23. S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," RFC-2212, September 1997.

24. J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC-2211, September 1997.

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

26. D. Ferrari, and D. Verma, "A Scheme for Real-Time Channel Establishment in Wide Area Networks," *IEEE JSAC*, Vol. 8, No. 3, pp. 368-379, April 1990.

27. D. Clark, "The Design Philosophy of the DARPA Internet Protocol," *ACM SIGCOMM '88*, August 1988.

28. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC-2475, December 1998.

29. J. Postel, "Service Mappings," RFC-795, September 1981.

30. K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC-2474, December 1998.

31. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC-2597, June 1999.

32. V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC-2598, June 1999.

33. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering over MPLS," RFC-2702, September 1999.

34. X. Xiao, A. Hannan, B. Bailey, and L. Ni, "Traffic Engineering with MPLS in the Internet," *IEEE Network Magazine*, March 2000.

35. P. Neelakanta, *ATM Telecommunication – Principles and Implementation*, CRC Press, Boca Raton, 2000.

36. R. Addie, J. Burgin, and S. Sutherland, "B-ISDN Protocol Architecture," *Proc. Globecom 88*, pp. 22.6.1, Florida, December 1988.

37. N. Anerousis, A. Lazar, "Virtual Path Control for ATM Networks with Call Level Quality of Service Guarantees," *IEEE/ACM Transaction on Networking*, Vol. 6, No.2, April 1998.

38. I. Chlamtac, A. Farago, and T. Zhang, "Optimizing the System of Virtual Path," *IEEE/ACM Trans, Networking*, Vol. 2, No.6, pp. 581-587, December 1994.

39. S. Ahn, R. Tsang, S-R. Tong, and D. Du, "Virtual Path Layout Design on ATM Networks," *Proc. IEEE INFOCOM '94*, pp.192-200, 1994.

40. B. Ryu, H. Ohsaki, M. Murata, and H. Miyahara, "Design Algorithm for Virtual Path Based ATM Networks," *IEICE Trans. Commun.*, Vol. E79-B, No.2 February 1996.

41. R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and its

Application to Bandwidth Allocation in High-speed Network," *IEEE J. Select. Areas in Commun.*, Vol.9, No.7, pp.968-981, September 1991.

42. O. Sharma, *Markovian Queues*, Ellis Horwood Limited, 1990.

43. M. Schwartz, *Broadband Integrated Networks*, Chapter 3, in Traffic Characterization for Broadband Services, Prentice Hall PTR, 1996.

44. O. Goldreich, *Introduction to Complexity Theory*, Cambridge University Press, 1999.

45. P. Jiyapanichkul, and J. Lambert, "Optimal Resource Management with Dynamic Virtual Paths in ATM Networks," *Proc. of IEEE International Conference on Communications*, ICC. V3, pp. 1728-1732, 1998.

46. K. Murakami, and H. Kim, "Optimal capacity and flow assignment for self-healing ATM networks based on line and end-to-end restoration," *IEEE/ACM Trans. on Networking*, 6(2), pp. 207–221, 1998.

47. Y. Xiong, and L. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. on Networking*; 7(1), pp. 98–110, 1999.

48. W. Grover, *Mesh-based Survivable Networks*, Prentice Hall PTR, 2003.

49. K. Kawahara, S. Nakazawa, T. Takine, and Y. Oie, "Performance Analysis of Layer3 Switch: Case of Flow-Driven Connection Setup," *IEICE Trans. Commun.*, Vol. E83-B, No. 2, February 2000.

50. C. Lin, and L. Sheng, "Integration of Scheduling Real-Time Traffic and Cell Loss Control for ATM Networks," *IEICE Trans. Commun.*, Vol. E85-B, No. 4, April 2002.

51. N. Giroux, and S. Ganti, *Quality of Service in ATM Networks*, Prentice Hall PTR, 1999.

52. R. Jain, *The Art of Computer Systems Performance Analysis*, Wylie, 1991.

53. A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *GAMS, A User's Guide*, GAMS Development Corporation, 1998.

54. Z. Wang and J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", *IEEE JSAC*, Sept. 1996.

55. H. R. Lewis and L. Denenberg, *Data Structures and Their Algorithms*, HarperCollins, New York, 1991.

Appendix A

Introduction to *General Algebraic Modeling System* (GAMS)

GAMS is a high-level modeling system for mathematical programming problems and is capable of solving large-scale linear and non-linear programming models [53]. The GAMS has incorporated ideas drawn from relational database theory and mathematical programming and has attempted to merge theses ideas to suit the needs of strategic modelers. Relational database theory provides a structured framework for developing general data organization and transformation capabilities. Mathematical programming provides a way of describing a problem and a variety of methods for solving it.

The GAMS model representation is in a form that can be easily read by people and by computers. The GAMS program itself is the documentation of the model. The GAMS system is designed so that models can be solved on different types of computers with no change.

GAMS programs consist of one or more statements (sentences) that define data structures, initial values, data modifications, and symbolic relationships (equations). Symbols must be declared as to type before they are used, and must have values assigned before they can be referenced in assignment statements.

In GAMS system, there are two most common ways of organizing programs. The first style places the data first, followed by the model and then the solution statements. In this style of organization, the sets are placed first. Then the data are specified with parameters, scalar, and table statements. Next the model is defined with the variable, equation declaration, equation definition, and model statement. Finally the model is solved and the results are displayed. The second style emphasizes the model by placing it before the data. This is a particularly useful order when the model may be solved repeatedly with different data sets. There is a separation between declaration and definition.

There are five basic GAMS data types and each symbol or identifier must be declared to belong to one of the following groups: sets, variables, parameters, equations, acronyms and models.

Sets are fundamental building blocks in any GAMS model. They allow the model to be succinctly stated and easily read. GAMS allows sets with up to 10 dimensions. In GAMS, a variable is the GAMS name for what are called 'decision variables' by industrial Operations Research practitioners. They are the entities whose values are generally unknown until after a model has been solved. A GAMS variable, like all other identifiers, must be declared before it is referenced.

In GAMS, equations are the names for the symbolic algebraic relationships that will be used to generate the constraints in the model. An equation must be declared before it can be used. In GAMS, all equations are collected into groups and labeled by using the model statement so that they can be solved. Once the model has been put together through the model statement, one can now attempt to solve it using the solve statement. GAMS itself does not solve the problem, but passes the problem definition to one of a number of separate solver programs. GAMS comes with a number of solvers that can be chosen by users.

A GAMS model will be first compiled, and then run by using one of a number of solvers.

Appendix B

Dijkstra's algorithm [48] is a well-known "Shortest Path" routing algorithm. Dijkstra's algorithm solves the problem of finding the shortest path from a point in a graph (the source) to a destination. Given G = (V, E), where V is a set of vertices and E is a set of edges. Dijkstra's algorithm keeps two sets of vertices:

S: the set of vertices whose shortest paths from the source have already been determined;

V-S: the remaining vertices.

The other data structures needed are:

d: an array of best estimates of shortest path to each vertex;

p_i: an array of predecessors for each vertex.

The basic steps are as follows:

1. Initialize d and p_i;
2. Set S to empty;
3. While there are still vertices in V-S,
   i. Sort the vertices in V-S according to the current best estimate of their distance from the source;
   ii. Add u, the closest vertex in V-S, to S;
   iii. Relax all the vertices still in V-S connected to u.

The relaxation process updates the costs of all the vertices, v, connected to a vertex, u, if we could improve the best estimate of the shortest path to v by including (u, v) in the path to v.

The pseudo code of Dijkstra's algorithm is as follows:

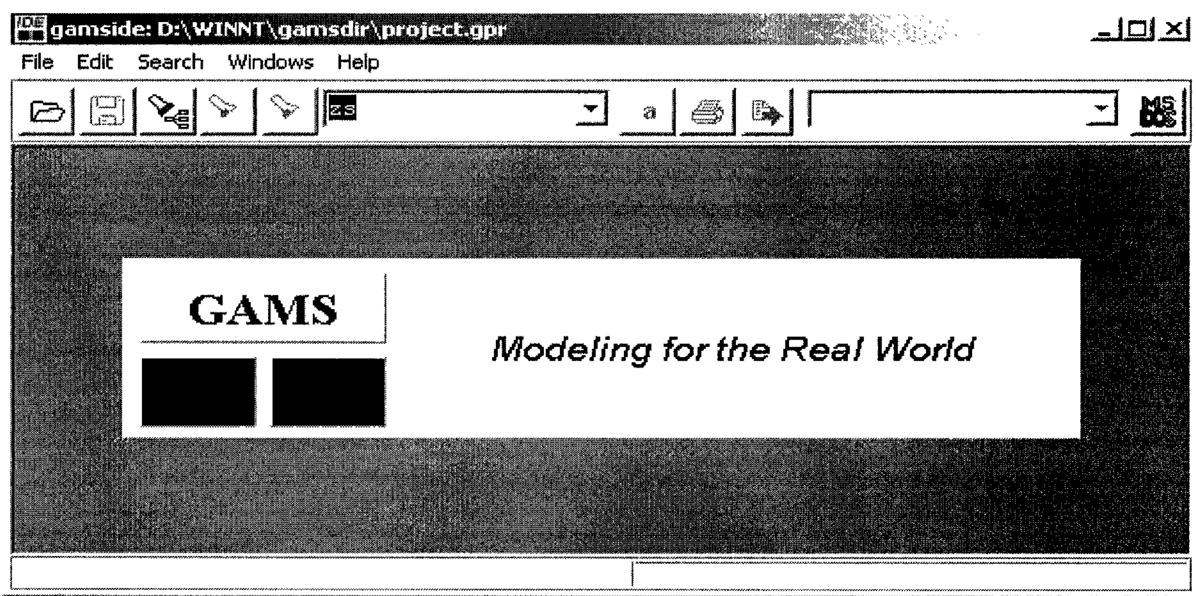
```
shortest_paths( Graph g, Node s )
   initialize_single_source( g, s )
   S := { 0 }                    /* Make S empty */
   Q := Vertices( g )            /* Put the vertices in a PQ */
   while not Empty(Q)
      u := ExtractCheapest( Q );
      AddNode( S, u );           /* Add u to S */
      for each vertex v in Adjacent( u )
         relax( u, v, w )

initialize_single_source( Graph g, Node s )
  for each vertex v in Vertices( g )
     g.d[v] := infinity
     g.pi[v] := nil
  g.d[s] := 0

relax( Node u, Node v, double w[ ][ ] )
   if d[v] > d[u] + w[u,v] then
     d[v] := d[u] + w[u,v]
     pi[v] := u
```

Appendix C

In GAMS IDE system, a typical working window is as follows:



Users have to generate a new GAMS file with extension "gams," and then they can edit the program in the above IDE environment. After finishing edit, they can compile it. After revising all errors, they can run it and get the output file with extension "lst." In the following section, we present the compiled codes for the LSP-CVP model and the LSP-L2 model.

GAMS Codes for the LSP-CVP Model and LSP-L2 Model

The following codes for the LSP-CVP model and the LSP-L2 model are based on the prototype network-2 (see Fig. 4-8).

LSP-CVP Model

Sets

p all paths in the network /1*50/
w all OD pairs in the network /1*25/
i all links in the network /1*35/

Parameters

d(p) the unit flow cost of path p in cases

```
/  1  27
   2  74
   3  62
   4  81
   5  52
   6  82
   7  44
   8  35
   9  72
   10 105
   11 72
   12 113
   13 86
   14 136
   15 90
   16 104
   17 38
   18 86
   19 83
   20 176
   21 86
   22 85
   23 58
   24 113
   25 68
   26 161
   27 81
   28 143
   29 101
   30 123
   31 98
   32 133
   33 122
   34 105
   35 96
   36 152
   37 119
   38 115
   39 71
   40 87
   41 45
   42 59
   43 104
   44 136
   45 103
   46 121
   47 127
   48 165
   49 158
   50 168 /
```

r(w) external traffic demand for every OD pair in cases

```
/  1  55
   2  76
   3  37
   4  17
   5  48
   6  60
```

```
        7  33
        8  69
        9  29
        10 46
        11 54
        12 40
        13 60
        14 63
        15 42
        16 35
        17 54
        18 29
        19 51
        20 42
        21 30
        22 51
        23 60
        24 41
        25 21 /

   c(i)   capacity for every link in cases
   /   1  200
       2  200
       3  179
       4  186
       5  200
       6  200
       7  200
       8  158
       9  200
       10 200
       11 200
       12 160
       13 200
       14 133
       15 200
       16 168
       17 200
       18 200
       19 184
       20 200
       21 200
       22 200
       23 200
       24 145
       25 200
       26 200
       27 200
       28 200
       29 200
       30 173
       31 200
       32 200
       33 160
       34 200
       35 200  /

Table pr(w,p) indicator used to indicate if a path p connects OD pair w
         1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
   1     110000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   2     001100000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   3     000011000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   4     000000110 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   5     000000001 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   6     000000000 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   7     000000000 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   8     000000000 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   9     000000000 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
10    000000000 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11    000000000 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
20    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
21    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
22    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
23    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
24    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
25    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;

Table pc(i,p) indicator used to indicate if a path p passes link i
        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
1     010000000 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
2     000001000 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0
3     001000000 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4     100100000 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
5     010001000 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0
6     000010000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0
7     000000000 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8     010101100 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1
9     100000010 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
10    000000001 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
11    010000001 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12    000000001 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
13    000000000 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
14    001000000 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15    000100000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1
16    000001100 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
17    000010010 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
18    000000000 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
19    000000000 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
20    000000000 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
21    000100000 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
22    000000000 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
23    000000000 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
24    000000000 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
25    000000000 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
26    000000000 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
27    000000000 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
28    000000000 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0
29    000000000 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1
30    000000000 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
31    000000000 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
32    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
33    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
34    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
35    000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1;

Variables
      x(p) virtual path flow
      z    total delay in network ;

Positive variable x;

Equations
      delay          define objective function
      demand(w)      satisfy demand of OD pair w
      capacity(i)    observe capacity limit at link i;
```

```
delay ..  z =e= sum(p, d(p)*x(p));

demand(w) ..  sum(p, pr(w,p)*x(p)) =e= r(w);

capacity(i) ..  sum(p, pc(i,p)*x(p)) =l= c(i);

Model network /all/;

Solve network using lp minimizing z;

Display x.l;

parameter lf(i) link flow on link i;
        lf(i) = sum(p, pc(i,p)*x.l(p));
Display lf;

parameter ratio(i) link utilization ratio;
        ratio(i) = lf(i)/c(i);
Display ratio;

parameter mean mean value of link utilization;
        mean = sum(i, ratio(i))/35;
Display mean;

parameter std standard deviation of link utilization;
        std = sqrt(sum(i, (ratio(i)-mean)*(ratio(i)-mean))/35);

Display std;
Display z.l;
```

LSP-L2 Model

```
Sets
       p  all paths in the network /1*50/
       w  all OD pairs in the network /1*25/
       i  all links in the network /1*35/

Parameters
       d(p)    the unit flow cost of path p in cases
       /  1  27
          2  74
          3  62
          4  81
          5  52
          6  82
          7  44
          8  35
          9  72
          10  105
          11  72
          12  113
          13  86
```

14 136
15 90
16 104
17 38
18 86
19 83
20 176
21 86
22 85
23 58
24 113
25 68
26 161
27 81
28 143
29 101
30 123
31 98
32 133
33 122
34 105
35 96
36 152
37 119
38 115
39 71
40 87
41 45
42 59
43 104
44 136
45 103
46 121
47 127
48 165
49 158
50 168 /

r(w) external traffic demand for every OD pair in cases
/ 1 55
2 76
3 37
4 17
5 48
6 60
7 33
8 69
9 29
10 46
11 54
12 40
13 60
14 63
15 42
16 35
17 54
18 29
19 51
20 42
21 30
22 51
23 60
24 41
25 21 /

c(i) capacity for every link in cases
/ 1 200
2 200
3 179

```
4  186
5  200
6  200
7  200
8  158
9  200
10  200
11  200
12  160
13  200
14  133
15  200
16  168
17  200
18  200
19  184
20  200
21  200
22  200
23  200
24  145
25  200
26  200
27  200
28  200
29  200
30  173
31  200
32  200
33  160
34  200
35  200  /
```

Table pr(w,p) indicator used to indicate if a path p connects OD pair w

```
         1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
     1   1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     2   0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     3   0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     4   0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     5   0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     6   0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     7   0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     8   0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     9   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    10   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    11   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    12   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    13   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    14   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    15   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    16   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    17   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    18   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    19   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
    20   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
    21   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
    22   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
    23   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
    24   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
    25   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
```

Table pc(i,p) indicator used to indicate if a path p passes link i

```
         1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
     1   0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
     2   0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0
     3   0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     4   1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
```

```
5    010001000 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0
6    000010000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0
7    000000000 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8    010101100 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1
9    100000010 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
10   000000001 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
11   010000001 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12   000000001 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
13   000000000 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
14   001000000 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15   000100000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1
16   000001100 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
17   000010010 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
18   000000000 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
19   000000000 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
20   000000000 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
21   000100000 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
22   000000000 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
23   000000000 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
24   000000000 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
25   000000000 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
26   000000000 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
27   000000000 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
28   000000000 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0
29   000000000 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1
30   000000000 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
31   000000000 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
32   000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
33   000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
34   000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
35   000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1;

Variables
     x(p) virtual path flow
     k(p) Layer 2 cut-through switching flow assignment
     n(p) virtual path capacity
     z    total delay in network ;

Positive variable x;
Positive variable k;
Positive variable n;

Equations
     delay           define objective function
     demand(w)       satisfy demand of OD pair w
     capacity(i)     observe capacity limit at link i
     pathlimit(p)    observe capacity limit at path p;

delay .. z =e= sum(p, d(p)*x(p)*(1-exp(k(p)*log(0.8)))/(1-exp(k(p)*log(0.8)+log(0.8))));

demand(w) ..  sum(p, pr(w,p)*x(p)) =e= r(w);

capacity(i) ..  sum(p, pc(i,p)*n(p)) =l= c(i);

pathlimit(p) ..  x(p)/n(p)-2*k(p)/(2*k(p)+6.908) =l= 0;

Model network /all/;

Solve network using nlp minimizing z;

Display x.l, k.l, n.l ;
```

```
parameter lf(i) link flow on link i;
        lf(i) = sum(p, pc(i,p)*x.l(p));
Display lf;

parameter ratio(i) link utilization ratio;
        ratio(i) = lf(i)/c(i);
Display ratio;

parameter mean mean value of link utilization;
        mean = sum(i, ratio(i))/35;
Display mean;

parameter std standard deviation of link utilization;
        std = sqrt(sum(i, (ratio(i)-mean)*(ratio(i)-mean))/35);

Display std;
Display z.l;
```