INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

NOTE TO USERS

This reproduction is the best copy available.

UMI®

Integrated Network Management System for MPLS-based Networks

By
Vandana Mandalika

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Science
in Information and Systems Science

School of Computer Science
Carleton University
Ottawa, Ontario, Canada

March 2005

Abstract

We architect an integrated network management system for Multi-Protocol Label Switching (MPLS) based network. The system provides a Graphical User Interface (GUI) front-end that can graphically display the topology and emulate an MPLS network. It allows the network manager to emulate Labeled Switched Paths (LSPs) before requesting MPLS signaling to actually set it up. A modified constraint-based routing algorithm is introduced that calculates alternate paths so that redundancy can be accomplished. In addition to the emulator, the system interacts directly with the routing component of the router. We provide a unique set of programming interfaces that directly accesses the Open Shortest Path First algorithm with Traffic Engineering (OSPF-TE) database to update the topology regularly. These unique interfaces are architected in such a way that they are router invariant and any third party routers can use them. In addition, the system provides a network management entity based on Command Line Interface (CLI) and web through which MPLS signaling modules such as with Resource Reservation Protocol with Traffic Engineering (RSVP-TE) can be directly invoked to set up a LSP. This novel method allows the network administrators to make intelligent decisions outside the router and send LSP set up commands only when they are sure about the network paths and their availability. This also empowers the network administrators to emulate and choose certain paths over others and to incorporate fault-tolerance and congestion management. Finally, we conduct performance analysis of this architecture and show that the overhead due to external calculation of LSPs are very minimal and do not add additional overhead to router resources.

Acknowledgments

I express my deep sense of gratitude to my supervisor Professor Sivarama Dandamudi for giving me an opportunity to work on this research topic and providing guidance and inspiration.

I sincerely thank Dr. Anand Srinivasan for his vision and guidance during the course of this research.

I take this opportunity to thank the team at Eion Inc., my family members and friends for their encouragement and support.

Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Full text
API	Application Programming Interface
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CBR	Constraint Based Routing
CIM	Common Information Model
CLI	Command Line Interface
CMIP	Common Management Information Protocol
CPU	Central Processing Unit
CSPF	Constrained Shortest Path First
DMTF	Desktop Management Task Force
FCAPS	Fault, Configuration, Accounting, Performance, and Security management
FDDI	Fiber Distributed Data Interface
FEC	Forward Equivalence Class
GLT	Graphic Layout Toolkit
GUI	Graphical User Interface
HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transfer Protocol
ID	Identifier
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IIS	Microsoft Internet Information Server
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IPX	Inter-network Packet Exchange
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
ITU-T	International Telecommunications Union Standardization Sector
JDMK	Java Dynamic Management Kit
JMX	Java Management eXtension
JNI	Java Native Interface
LAN	Local Area Network
LS	Link State
LSDB	Link State Data-Base
LSA	Link State Advertisement
LSP	Labeled Switched Path
LSR	Label Switched Router
MB	Mega Bytes
MPLS	Multi-Protocol Label Switching
ms	Milli Seconds
NE	Network Element

Acronym	Full Text
NM	Network Management
NP	Non-deterministic Polynomial
NMS	Network Management System
OC	Optical Carrier
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPF-TE	Open Shortest Path First - Traffic Engineering
QoS	Quality of Service
PBNM	Policy Based Network Management
RIP	Routing Information Protocol
RMON	Remote Monitoring
RPC	Remote Procedure Call
RSVP	Resource Reservation Setup Protocol
RSVP-TE	Resource Reservation Setup Protocol - Traffic Engineering
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SPF	Shortest Path First
SQL	Standard Query Language
TCP	Transmission Control Protocol
TE	Traffic Engineering
TMN	Telecommunications Management Network
TLV	Type/Length/Value
ToS	Type of Service
UDP	User Datagram Protocol
VPN	Virtual Private Network
WBEM	Web-Based Enterprise Management
WMI	Windows Management Instrumentation
XML	eXtensible Markup Language
XSLT	XML Style-sheet Language Transformations

Chapter 1

Introduction

Till recently, the mode of communication was restricted to asynchronous data traffic like electronic mail and file transfers. But, with the growing demand and publicity of multimedia, the transmissions need to support real-time traffic like video and voice. The two requirements of real-time traffic are time bounded service and bandwidth guarantee. Internet Protocol (IP) is a well-known layer 3 protocol that carries packets hop-by-hop to its destination. Since IP does not guarantee predictable path for packet transfer between a source and destination, another protocol was required to guarantee such predictability. Multi-Protocol Label Switching (MPLS) was introduced between layer 2 and 3 of OSI protocol stack to add the connection-oriented feature over connection less medium. MPLS guarantees bandwidth and other constraints on a label switched path between the source and the destination, which predicts the packet route in advance [1].

A typical carrier or service provider has thousands of nodes (or routers) in an IP-based network. These nodes are connected among each other to form a complex topology supporting various protocols to cater different customer needs. With the introduction of MPLS to support real-time traffic over IP networks, managing all the connections has become very challenging. The network administrators require a system that would make their work efficient in providing a way to control the network from a single place. The system should provide with a front-end Graphical User Interface (GUI) to manage the network seamlessly from any geographical location. The front-end should also provide

with a mechanism to login to any router and configure the router at any time. The system should also provide with various network management interfaces to the router so that the management could either be done manually or automatically. The system should interface with the router at all time to obtain the latest topology and help the network manager to run "what-if" scenarios regarding failures. The same topology should be useful for network manager to create static paths with various constraints using a basket of optimization algorithms. Moreover, many of the applications are time sensitive, demanding the optimal routes as against shortest routes to reach the destination. Traffic Engineering (TE) aspect of Network Management attempts to cater such needs. Today, such an integrated network management system is not available to control routers remotely and intelligently. In this thesis, a system is designed, developed, and tested that would allow the network administrators to manage the system remotely with built-in intelligence in the system.

In the following sections of this chapter, an introduction to network management and traffic engineering are presented.

1.1 Network Management System

Although network management may mean different to different people, a well-accepted concept of it is monitoring and controlling the network. Typical network management system consists of a set of *network elements* (NE) or nodes, and a N*etwork Management System* (NMS) that manages the network elements. Figure 1 describes a typical physical network management system. In this diagram, a few network elements are shown involving a few servers running different applications, a print server, a router and an

Remote Monitoring (RMON) probe inside the network. The network management workstation collects the necessary information, using which the network manager monitors and controls the network [2].

In the data communications world, *routers* are the most important network elements that are managed. For that reason, in this thesis, network element, node and router are sometimes used interchangeably. Similarly, the network management system is also sometimes called *network manager*. Whereas, the person managing the network is termed as *network administrator* or network operator.

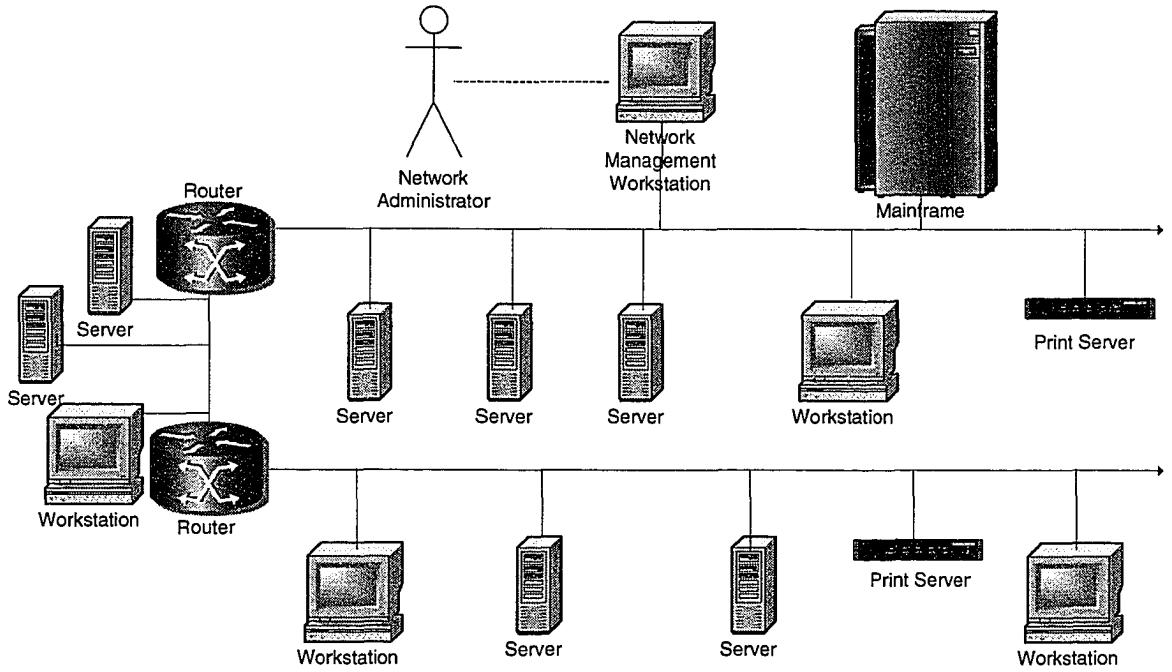


Figure 1: Network Management System - physical view

Each network element has a set of objects to be managed by the NMS, called managed objects or *managed entities*. The information about the managed entities from managed elements is collected using a *management protocol*. On the managed element side, a

software component known as *agent* interacts with the network manager to exchange the *management information.* Figure 2 describes the logical view of a network management system.

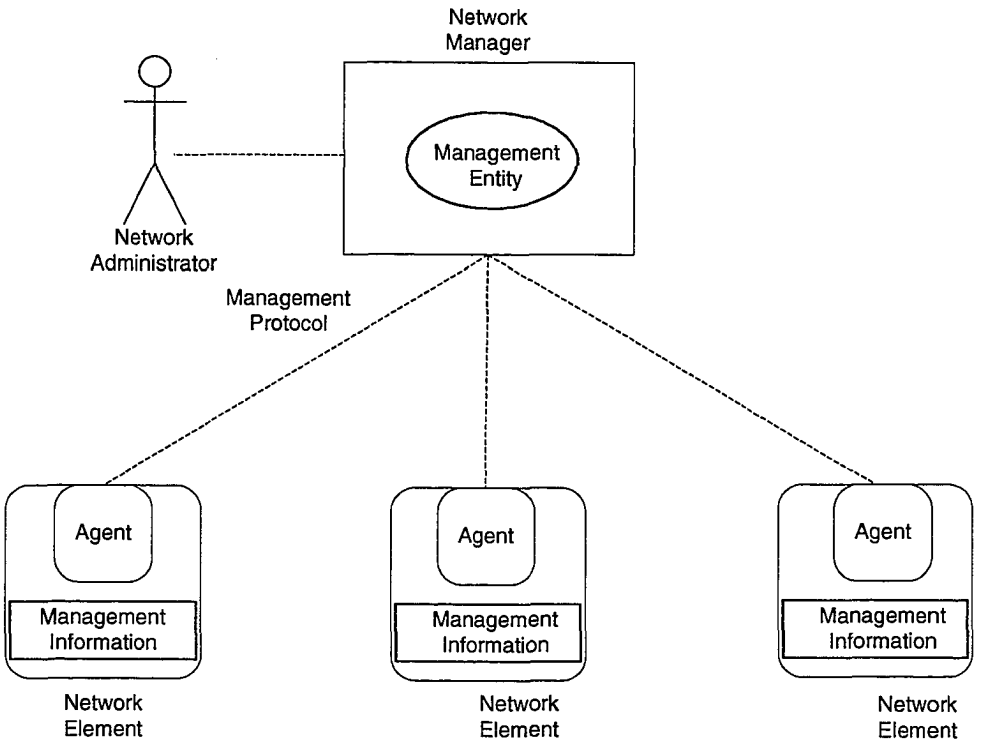


Figure 2: Network Management System – logical view

1.2 Network Management Functional Areas

The International Organization for Standardization (ISO) identified 5 functional areas of network management for Open Systems Interconnection (OSI) [3-4]. Although they are defined for OSI, they are representative of different aspects of network management. The functional areas defined by ISO that are famously known as FCAPS are:

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management

Fault management deals with detecting, isolating, and correcting abnormal operation in the network. Fault management may also involve alarm correlation and diagnostic testing to pin point the location and reason for the fault and take appropriate action to correct the fault.

Configuration management identifies and gathers the network information for the purpose of provisioning the resources and services. This involves initialization and graceful shutdown of the network, adding and updating relationship among various network components and their status is also considered as part of configuration management.

Accounting management enables collecting the usage of network resources and charges to be established for their use. The network accounting information may be collected and charged at different levels like department, division, or organization. Although internal charging to department levels is not always employed, network manager would like to keep track of the use of network resources to determine the network abuse, to make better use of the network, and plan for network growth by understanding the user activity.

Performance management provides the facilities to evaluate the behavior of network elements and the effectiveness of communication. This includes functions to collect network statistics, and applying control to prevent, rather than react to, traffic congestion. When certain performance parameters exceed the thresholds, events are raised to help assist in analyzing the network behavior. Performance management also includes monitoring and controlling the Quality of Service (QoS) in a network.

Security management involves security of management and management of security. It is important to securely communicate the management information. Security threats and services required to overcome the threats are defined using access control, authentication and encryption. For example, encrypting customer credit card information using public key cryptography is securing management information, whereas defining procedures to manage the key such as measures to be taken if the private key is compromised is management of security.

Telecommunications Management Network, as defined by International Telecommunications Union Standardization Sector (ITU-T), is a well-known architectural model in network management domain and is closely related to OSI management.

1.3 Telecommunications Management Network

Telecommunications Management Network (TMN) is conceptually a separate network that interfaces a telecommunication network at different points [41]. TMN recommendation M.3010 defines general TMN management concepts and several

management architectures at different levels of abstraction. Five hierarchical layers that separate the responsibilities of network management aspects are defined as follows.

- Network element layer – this layer is not a management layer but this is where the network elements reside.
- Element management layer – this layer is directly responsible to manage the network elements. Tasks in this layer include detection of equipment errors and logging of statistical data.
- Network management layer – this layer manages interaction between multiple pieces of equipment. Tasks in this layer include monitoring link utilization and optimizing network performance.
- Service management layer – this layer deals with management aspects that may be observable by users of the telecommunication network. Tasks in this layer include quality of service and accounting.
- Business management layer – this layer is concerned with the management of whole enterprise with a broad scope that sets the goals for running the network.

The concepts in OSI and TMN set guidelines for the area of network management. To manage a network, network manager typically uses configuration, fault and performance management tools. Not many network management products provide security and accounting management features. In addition to the tools in traditional functional areas of network management discussed above, a network manager needs more tools to manage the network. Notably, with the evolution of MPLS, setting up of the routes, capacity planning, ensuring that the QoS parameters are met in the network borrow the principles

from traffic engineering. In the next section, traffic engineering and MPLS concepts are presented.

1.4 Traffic Engineering and MPLS

Efficiency and cost structure in a network operation is determined by how the available bandwidth is utilized [1]. Efficient use of the bandwidth involves avoiding the situation where some parts of the networks are congested while the others are underutilized. An approach to solve this traffic-engineering problem is to specify the routes in the network to efficiently utilize the network links [5-8].

Traditional IP routing determines the route based on the destination of the packets. Algorithms like minimum hop routing result in using the same path for given source and destination. One way to change the routing is by modifying the metrics used by the routing protocols. By merely changing the cost on the links, the situation between congested link and under utilized links, the congestion migrates to the new path with low cost link but the problem will not be solved. Features like equal-cost and multi-path capabilities provided by OSPF or IS-IS protocols may appear to solve this problem. However, if there are links with different bandwidths, it can be shown that the links will still be unevenly utilized [1]. One approach to solve this problem is to use the available link bandwidth as a parameter in addition to certain scalar metric (like distance or hops) in deciding the route.

MPLS is a standard from IETF that can be used to solve the traffic engineering problem found in traditional IP networks. MPLS can be used between any of the OSI network

layer protocols IPv4, IPv6, AppleTalk, IPX, and OSI link layer protocols Ethernet, FDDI, ATM, Frame Relay, Point-to-Point. Figure 3 describes MPLS operation in an IP network that may be running network layer protocol of IPv4 and link layer protocol of Ethernet.

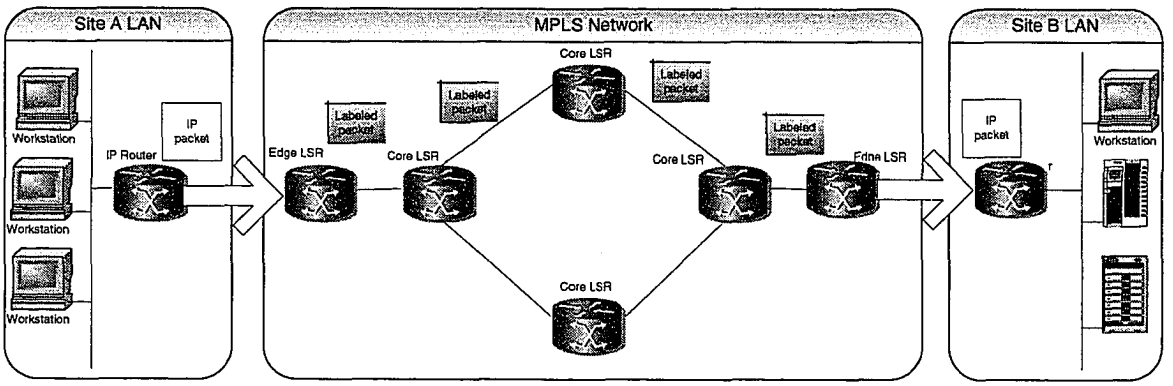


Figure 3: MPLS with an IP network

MPLS network comprises of Label Switched Routers (LSRs) that can switch and route packets based on the label that is appended to each packet. Labels define a flow of packets between two end points. In case of multicast, labels define flow of packets between a source endpoint and a multicast group of destination endpoints. For each distinct flow, called Forward Equivalence Class (FEC), a specific path through the network of LSRs is defined. For each FEC, its traffic characterization is defined as a set of Quality of Service (QoS) parameters for that flow. LSRs do not examine or process the IP header. The packets are forwarded based on the label value, making the forwarding process simpler than IP router.

For a given FEC, before routing and delivery of packets based on the QoS parameters, a LSP must be established. QoS parameters determine the resources to be committed to the path and queuing and discard policies at each LSR for packets in the FEC.

When a packet enters an MPLS domain at ingress edge LSR, it is processed to determine the QoS parameters and the network layer services it requires. LSR assigns this packet to a FEC, associated to a LSP and appends label to the packet and forwards the packet. If no LSP exists for this FEC, ingress edge LSR works with other LSRs to define a new LSP. As the packet goes through the other non-edge LSRs, incoming label is removed and outgoing label is attached. The packet is forwarded to the next LSR in the LSP. At the egress edge LSR, the label is stripped; IP packet header is read and forwarded towards its destination.

Route selection is defining LSP for an FEC. Each LSR on the LSP must assign a label to the LSP to be used to recognize incoming packets that belong to the FEC. LSR must inform all potential upstream nodes that will send packets for this FEC to this LSR of the label assigned by this LSR to this FEC so that those nodes label the packets. LSR also should learn the next hop for this LSP and learn the label that the downstream node has assigned to this FEC. This will enable this LSR to map an incoming label to an outgoing label.

The decision to switch to the next hop is not necessarily based on the destination IP address as is usually done in IP based routing. Any of the policies can be used to determine the complete end-to-end path of a flow. When the path is determined based on the bandwidth available along the route, this would result in efficient use of the network links. Although the source-based routing can be used in an IP based network, MPLS achieves performance gain by bundling multiple flows in an LSP.

1.5 Organization of the Thesis

Chapter 2 introduces network management interfaces using command line interface and web-based network management. A survey of related products in network management and traffic engineering is also presented.

Chapter 3 describes the problem addressed in this thesis. It also covers how this problem considered important and how it compares with the existing approaches.

Chapter 4 provides architecture and design principles adopted in this thesis.

Chapter 5 gives implementation details of the system that was designed and developed to demonstrate the concepts.

Chapter 6 presents performance analysis.

Chapter 7 summarizes the thesis and suggests future work.

Chapter 2

Previous Work

In this chapter, command line interface to configure network element, and accessing network manager using web-based interface are discussed. A review of the literature in network management and traffic engineering areas is provided. The network management features and products, especially for managing IP-based MPLS networks are the subject of this chapter.

2.1 Command Line Interface

Network elements, specifically routers have a lot of management information to configure when deployed in the field to suit the network. Most common form of managing the router is using the Command Line Interface (CLI). Even after deployment, the CLI is used for configuration management of the router.

Traditionally, Unix users are very much used to running commands on operating system shell. Such users prefer CLI instead of running a command in GUI. Also, CLI commands come in very handy to sequentially run a set of commands in a batch file.

Cisco is one of the first companies to introduce the concept of CLI to configure its routers [9]. A typical Cisco style configuration of routers using CLI is done in multiple levels. At each level, there is a context in which some commands can be run. For example, to shutdown a fast Ethernet interface on port 5, the following sequence of commands is run in different contexts.

At the operating system command prompt, *config* command is run, which takes into the *config* mode.

hostname# **configure**

hostname(config)#

Now, to configure fast Ethernet interface on port 5, *interface* command is used with arguments *fastethernet* and the port number. That switches the context to *interface configuration.*

hostname(config)# ***interface fastethernet 0/5***

hostname(config-if)#

The last step in this sequence is to shutdown the interface using the command *shutdown.*

hostname(config-if)# ***shutdown***

In addition, there are some CLI commands that can be executed like absolute path from anywhere. For example, the following command is run to display the status and statistics of fast Ethernet interface on port 1.

hostname# ***show interfaces fastethernet 0/1***

2.2 Web-based Management

Network manager is usually a standalone application. To use the network manager, usually the network administrator logs into the node where network manager is installed. Alternatively, a thin-client is made available to connect to the network manager and

network operator has to install the thin-client and connect to the network manager. Both of these options are not quite attractive. It would be better if the network operator can open a browser and connect to the network manager from anywhere in the network. There are a few web-based network management products out in the market doing similar operation. One of them is *DR.-Web Manager* from SNMP Research [10]. As described in Figure 4, a web server resides in the network manager and allows connections from any web browser connected using HTTP [11].

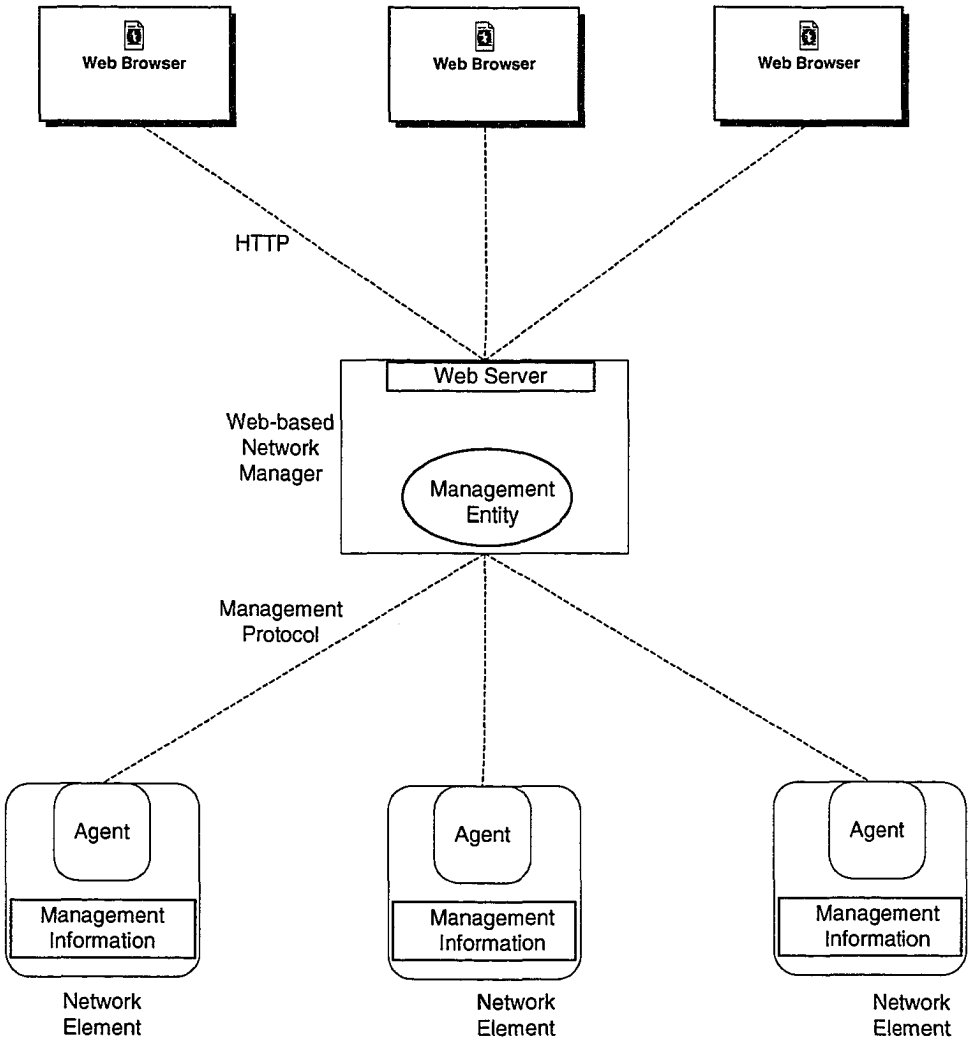


Figure 4: Web-based Network Management

2.3 A Survey of NM and TE Products

There are quite a few generic network management products in the market. Some of them are generic enough while others are specific to the network elements being managed. In this section, some of the most used network management and traffic engineering products and technologies for IP and MPLS based networks that are relevant to the work done in this thesis are discussed.

OpenView from Hewlett Packard is a network services management solution suite [12]. The solution allows continuous monitoring, reporting, and troubleshooting, and automated response capabilities. The solution is targeted for services such as MPLS and IP Telephony. OpenView uses Simple Network Management Protocol (SNMP) as the base network management protocol. For a typical application that involves management of MPLS networks, the recommended set of tools from OpenView suite is: Network Node Manager, Performance Insight, Network Management Smart Plug-in for MPLS, Performance Insight Report Pack for Traffic Profile. HP Network Services Management solution for MPLS provides the tools to:

- Discover and monitor elements, relationships, dependencies, traffic routing and services.
- Tie physical faults to a specific VPN and its associated customer.
- Predict and prioritize risks to services and customers.
- Prioritize the outstanding problems and react to locate the source of a problem.
- Measure MPLS VPN traffic and QoS.
- Meet and prove SLAs.

- Assets and capacity planning provide input on bandwidth buy-lease decisions.
- Generate custom reports to management or customers.

Tivoli NetView from IBM is a suite of network management products that cover a wide range of applications and network elements [13]. IBM Tivoli NetView extends network management functionality to ensure availability of critical business systems and to provide rapid resolution of problems. This suite is available in multiple Windows and Unix platforms for monitoring and controlling enterprise-wide multiprotocol network from a single console. NetView uses SNMP to manage the devices. Web-based network management feature is also available in this product. IBM Tivoli products closely related to the work discussed in this thesis are IBM Tivoli Monitoring for Network Performance, and IBM Tivoli Configuration Manager. IBM Tivoli network management products provide the tools to:

- Display network topologies.
- Groups network devices into collections for better manageability.
- Monitors network, gathers performance data.
- Manages events and SNMP traps.
- Correlates events and identifies root cause of network failures.
- Integrates with leading vendors such as CiscoWorks2000.
- Maintains device inventory for asset management.
- Reports on network trends and analysis.

VitalNet from Lucent is a SNMP based product that is used to measure, analyze and report on packet data and voice networks performance [14]. The performance data is

stored in a commercial relational database system and accessible using SQL interface. Specific features of this product include:

- Automatic discovery of network SNMP and RMON devices.
- Support for multiple devices from various vendors.
- Identifies performance problems.
- Verifies QoS delivery and SLA compliance
- Real-time surveillance.
- Real-time displays.
- Versatile thresholding.
- Event analysis.
- Event notification.
- Use trending data for capacity planning.
- Browser-based interface.

Policy Based Network Management (PBNM) is a class of solutions that attempts to manage QoS and security in distributed networks [15-17]. *Service Activator* of Orchestream (now part of Metasolv) is a policy based network management product that is used to create applications with QoS [18]. The product allows providers of virtual private networks (VPNs) to provide QoS guarantees between different points on the network. These guarantees ensure predictable end-to-end performance over IP-networks. Orchestream Service Activator provides the following features:

- The device information is stored at discovery time.

- Network configuration changes are accomplished by using transactions. These transactions can be scheduled and can also be backed out later.
- Allows users to optimize performance of the network
- Proactively manage service level agreements
- Includes roles, access groups, user IDs, sites, and per-hop behavior groups. Devices, interfaces and virtual/sub-interfaces are given roles.

Formulator from Gold Wire Technology (now part of *R-Series* from Intelliden) is also a policy based network management product that is used to collect, archive, compare and audit multi-vendor infrastructure configurations [19]. Formulator's Policy management is rule-centric and the rules are organized into a hierarchy. Formulator offers the following features:

- The product assures the integrity of the infrastructure by providing centralized access control over network devices
- Capturing audit trails of all activity and changes.

Redcell suite of products from Dorado Software offers performance and fault management in addition to policy management [20]. The salient features of this product are:

- The product can auto-discover SNMP devices.
- It can define QoS and access policies.
- Canned policies like Olympic series (Gold, Silver, and Bronze) service types are defined.

- Service management suite can create and manage service level agreements (SLAs) and link faults.

Many telecommunications network management products use Common Management Information Protocol (CMIP) that was developed by ISO. *Solstice Enterprise Manager* from Sun Microsystems is one of the products that uses CMIP to manage networks [21]. The suite can be used for rapid service creation and management of telecom networks and underlying equipment running Sun Solaris operating system on SPARC platform. The product allows standards-based and proprietary devices to be managed from common infrastructure.

Distributed Management Task Force (DMTF) developed Common Information Model (CIM) as a standard information model for describing management data about devices, networks, systems and applications of computer system [22]. CIM also allows vendor specific extensions. DMTF also defined the communication model to be Web-Based Enterprise Management (WBEM) that defines CIM operations as an interface to the model, an XML representation and a mapping to transport the CIM operations with the XML CIM model data using HTTP. Microsoft's *SMS* is a comprehensive solution for change and configuration management for Microsoft platform to provide relevant software upgrades [23]. SMS supports CIM/WBEM framework. Windows Management Instrumentation (WMI) is Microsoft's management infrastructure and is based on CIM model [24]. WMI includes components like SNMP providers to interoperate with different management protocol stacks.

XML based network management has also drawn significant attention and is used to manage some products [25-26]. XML syntax can be used to describe complex data structures with ease. Using XML-RPC to exchange data between network manager and the network element for carrying out the operations serves as the management protocol. XML Style-sheet Language Translation (XSLT) can be used to convert it to and from different formats.

Java Management Extension (JMX) from Sun Microsystems is an instrumentation toolkit for Java based products that provides a management architecture and API set to allow any Java technology based or accessible resource to be inherently manageable [27]. Sun Microsystems' *JDMK* includes a JMX implementation and a simple management console to support it. Several network management products like IBM *Tivoli* can interoperate with JMX to manage the application or the network element.

2.4 Comparison of NM and TE Products

From the network management technologies perspective, most of the device and network management products are using SNMP as the base management protocol. Telecommunications network management products use CMIP as the base management protocol. CIM/WBEM, XML and JMX are the upcoming management technologies.

As it can be seen from the available products, network management is a pretty mature domain where as, traffic engineering is still developing momentum. Most of the network management products are proprietary. Proprietary solutions compete on management

infrastructure and capabilities, services for installation and customization, and products and applications that can be managed.

Some of the generic architectures either need a lot of customization to suit the network being managed and/or are costly for a small network solution. Policy based network management products using traffic engineering features are very specialized. The work done in this thesis compares to some of the problems addressed by the network management, policy management and traffic engineering products.

Table 1 captures the comparison of the products in NM and TE area by feature.

#	Feature	HP OpenView	IBM Tivoli	Lucent VitalNet	Orchestream Service Activator	Gold Wire Formulator	Dorado Redcell
1	Fault management	√	√	√			√
2	Performance management			√	√		
3	Configuration management	√	√	√	√		√
4	Security management				√	√	
5	Accounting management					√	
6	SNMP	√	√	√			√
7	CLI						
8	Web-based interface			√			
9	Measure QoS	√		√	√		√
10	Support SLAs	√		√	√		√
11	Capacity planning	√		√			
12	Generate reports	√	√	√			

Table 1: Comparison of NM and TE products

Chapter 3

Problem Statement

3.1 Problem Addressed

In this thesis, the following problems are addressed:

- A new architecture for an integrated network management system to solve the complex problem of managing a MPLS network with minimum performance hit to the MPLS routers.
- Design and develop constraint-based routing as part of network manager for path calculation with constraints. This minimizes the use of router resources by separating complex path calculation functionality out of router and to provide more control to network administrators as shown in Figure 5.

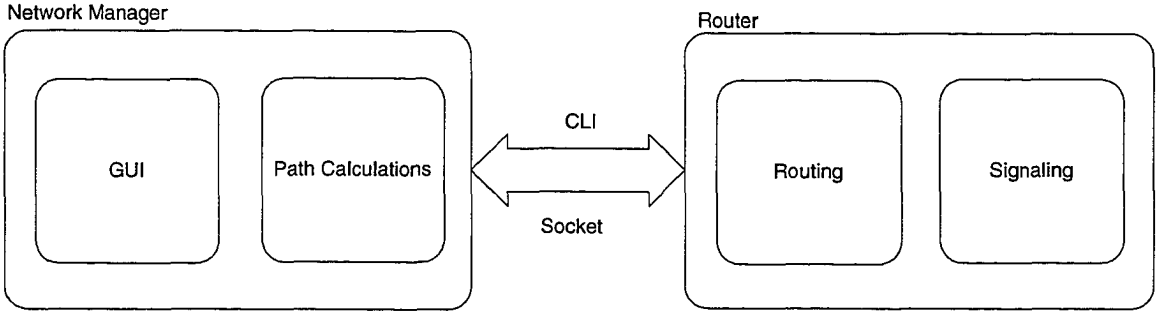


Figure 5: Separation of path calculation

- Integrate and create a common network management system that provides a platform to automatically communicate with routing module within a router and obtain network topology and graphically display the topology.
- Allow network administrators to do "what-if" scenarios on the topology, and make intelligent decisions on which primary path to take and which alternative path to keep. Emulate various topologies and provide redundant path for path calculation purposes.
- Invoke MPLS router CLI to setup LSPs from network manager.
- Create common APIs that can be used on any third party routers – removing the dependency on path calculation modules inside the routers and allowing to make intelligent decisions in network management systems.

3.2 Problem Details

As the real-time traffic on IP networks is increasing enormously, it is very important to have an efficient protocol that increases the packet forwarding performance, maximizes the resource usage, and optimizes the traffic flow to provide service differentiation. Addition of MPLS protocol to IP networks gives all the needed features mentioned above. However, the MPLS networks pose a great challenge to the network managers for managing complex topologies in which there are numerous nodes [1][28-29].

The nodes of a network need to communicate with each other in order to update their status to all other nodes in that network. In order to achieve this, OSPF protocol is used for communication between routers. OSPF is used for communicating the status of a link

in a router to its neighbors, propagation of the link state database to neighbors and calculation of the topology.

However, OSPF protocol does not carry the TE parameters, TE information is overlaid on top of the OSPF. IETF OSPF-TE protocol specifies the standard for exchanging TE parameters among the nodes [33].

A typical router has several functionalities that are traditionally managed under the umbrella of OSI FCAPS (fault, configuration, accounting, performance and security). Managing an MPLS-based router that has OSPF-TE functionality is complex. With stringent QoS guarantees required for real-time applications, mere FCAPS based management is insufficient. Network administrators are expected to be more proactive in analyzing the available paths and suitably set them. The established SLAs have to be guaranteed by demonstrating that the service is delivered. There is also a need to provide capacity planning.

Networks are dynamic entities, and having the latest network topology is a constant concern. Any router can become faulty, and router interfaces can become congested. The network administrator needs to know the network topology information. It is very important to have a user interface, which displays the network topology with all relevant information about nodes and links. The information about each node and edge needs to be stored in a persistent database. In order to have maximum flexibility in the management of such a box, different management tools like CLI, and web-based front-end need to be easily accessed from a single integrated network management system. All of this expands the scope of network management. Such a single integrated network

management system is not available in the market. Hence in this work, a useful, network administrator friendly integrated network management system is proposed and developed.

This thesis work was started with a network manager in a standalone simulation mode running in Windows platform using the Java programming language. Simulated network topology is read and interpreted from a static text file. The link bandwidth displayed on the network topology was assumed to be same on both directions. Network topology information for each node and each edge was then stored in Access database. Using Tom Sawyer software, the network topology of an area was drawn with the status of nodes and links as active, inactive and congested. The network topology information can also be displayed in tabular form as a list of edges. Each entry in the table displays node adjacencies and status about nodes and links. Node redundant and edge redundant routes can be computed for any pair of source and destination using Dijkstra's shortest path algorithm developed in C programming language. The network topology graph is completely redrawn on each refresh. On the other hand, the MPLS-based routers are capable of setting the LSPs using RSVP-TE.

To assist MPLS-based network managers, in this thesis work, the existing simulation system is enhanced to interact with routers in real-time. The router responds to the network manager request by gathering network configuration data from the OSPF-TE link state database.

On receiving the network information from router, the network manager displays current network topology as requested by network administrator. In addition to displaying the

topology, it displays the node status (up or down), link status (up or down), and bandwidth available on the link in both directions. Whenever there is a request to refresh the network topology, the latest update from the network is compared with the previous topology and the differences are displayed as status change in nodes or links. This also updates the link bandwidth information in each direction.

To enable the network administrator to analyze what-if scenarios such as finding an optimum route from one node to another using the current network topology, Constraint Shortest Path First (CSPF) based routing algorithm is developed on the network manager side. To allow the network administrator to manage the network remotely from a single location, an option is provided to remotely login into the node and modify or update the node with CLI commands, for example, to set up the LSP path between the source node and the destination node. Taking advantage of the web-based evolution, capability is presented to manage the network by accessing the network manager from anywhere using a web browser.

3.3 Problem Importance

To manage the network efficiently, network administrator needs the latest configuration of the network. Making use of OSPF-TE parameters help analyze the network topology in a greater detail including the bandwidth available on each link in both directions. This is the first step that lays foundation for the other aspects addressed in this thesis.

Network managers often need to analyze the what-if scenarios for setting up different LSP paths in the MPLS network with constraints like minimum bandwidth required along

the path, node and link outages. However, trying them out in the real network by reserving and backing out is expensive and wastes scarce network resources. Thus, what-if scenario analysis is carried out outside the router, at network manager. Also, using this approach, no network resources are reserved till the real route to be set is identified. A constraint-based routing algorithm is developed at the network manager using current network topology to compute the node- and edge-redundant paths.

To provide integrated network management from a single network manager, it should be possible to access each of the routers in the network. For example, when what-if scenario are analyzed and the network administrator is ready to set the path, a handy way to do that is by invoking router Command Line Interface (CLI) right from the router icon in the network topology displayed at the network manager. Such a network administrator friendly interface is developed in this work.

To access the integrated network manager from any host in the network without even having to install a thin client, a web-based network management user interface is developed. To achieve this, a web-based server is deployed on the network manager. The web server accepts connection from a web browser running on any host in the network.

3.4 Comparison with the Existing Approaches

Telecommunication Management Network [41] specifies the following logical layers: network element, element management, network management, service management, and business management. Work in this thesis overlaps with the network element management and network management layers of TMN. Invoking router CLI carries out

element management layer functionality. Network management layer functionality is achieved by displaying the network topology and indication of network faults. Also, creation of dedicated paths to support QoS requirements falls under the network management layer functionality. Most of the network management products either implement element management layer functionality or that of network management layer functionality.

Most of the available network management products typically use heavy weight network management protocol, resulting in expensive solution, which are not affordable for small network installations. In this thesis, an inexpensive approach to gathering network data is used.

No product is known to be available in the market that analyzes what-if scenarios without using the network resources. In this thesis, such functionality is built-in to the network manager. Integrated network management is achieved by remotely managing routers using a CLI. As found in very few products, web-based network management is developed.

Chapter 4

Architecture and Design

As in every information system life cycle, architecture and design of network management system open a variety of options and considerations. That involves where to collect the data from, how to gather the information, in what format to exchanging the information, where to store the information, how to display it to the user, how to update, and how to manage the information. In this chapter, all the architectural issues are considered and the design options made are discussed.

4.1 Gathering Network Configuration

In gathering the network configuration, there are several aspects.

- Location of network management information.
- Communication between network element and network manager.
- Source of network topology information.
- Network topology information update.

Each of these architectural options is discussed in the following subsections.

4.1.1 Location of Network Management Intelligence

In managing the network element, there are two options. First approach is to have the management information on the network element and directly managing it. Second

approach is to gather the network element information to another entity, outside the network element and managing from there. There are pros and cons to each of the approaches.

Advantages of having network management intelligence within the router are:

- The topology information is latest as it is instantaneously available on the router.
- The algorithms that are run on the router can also be used to find a path between a source node and a destination node, for the "what-if" scenarios.
- This approach eliminates having a separate entity to maintain the router intelligence.

On the other hand, advantages of having network management intelligence outside the router are:

- The router's valuable resources are not drawn away from its main responsibilities.
- "What-if" scenarios can be analyzed under different conditions like node failure and link failure.
- Router uses a restricted set of algorithms to compute a path. The network manager need not be restricted to the algorithms used by the router. These "what-if" scenarios can also be used to create static paths with various constraints using a basket of optimization algorithms.
- Independent network manager gives the additional advantage that it can be non-platform specific as it can be run from anywhere.

In favor of analyzing the what-if scenarios without consuming router resources, having network element intelligence outside in a network manager is opted.

4.1.2 Communication Between Router and Network Manager

In a typical network management scenario where the network manager is residing outside to control the network element, the communication protocol defines the semantics of the information exchanged between them. Two options, namely, SNMP and simple socket-based communication are considered. The analysis of the options is described below.

SNMP is one of the very dominant management protocols in the Internet community [30]. Three versions of this protocol suite were released. The first version provided simple facility for management that is easy to implement and uses minimum processing resources. Version 2 of SNMP includes functional enhancements to the first version and improves the efficiency of communication by using bulk transport mechanisms and enhanced error handling. While the first two versions of SNMP only provided community-based form of security (read-only and read-write attributes), version 3 significantly enhanced that by adding several encryption mechanisms, authentication strategy and customizable permission levels for different users. SNMP uses UDP/IP for communicating management information. The network manager (called as SNMP manager) communicates with the network element (called as SNMP agent) in three ways:

1. Get requests are used by the SNMP manager to obtain information from the SNMP agent. First element is received by sending a GET. Subsequent entries of a table are

retrieved by GET-NEXT, requests. BULK-GET (available beyond since Version 2) can be used to retrieve complete table.

2. Set requests are used by the SNMP manager to configure managed entities on the SNMP agent. BULK-SET can be used to configure multiple parameters.
3. Notifications (also called as traps or inform messages depending on the version used) are asynchronously sent by the SNMP agent to the SNMP manager to inform abnormal events. Network administrator configures the SNMP agent for the notifications that should be sent.

On the other hand, a socket-based connection between the network element and the network manager is a simple mode of communication. Socket-based communication can be reliably used between any devices running TCP/IP protocol suite, using either stream-based connection (TCP). However, the communicating entities need to define semantics to interpret the information that is exchanged.

To keep the communication mechanism simple and lightweight, socket-based connection between the network element and the network manager is chosen.

4.1.3 Source of Network Topology Information

Network topology from a router can be gathered from multiple places [31]. The router gathers the intra-network topology information using the routing protocols RIP and OSPF, and the inter-network topology using IS-IS and BGP. Information about bandwidth on the links is useful to route the traffic based on traffic engineering concepts. The network topology can also be collected from the OSPF-TE protocol, which has

bandwidth information as well. It is proposed to have the topology information gathered from different routing protocols as shown in Figure 6. For network management involving traffic engineering, bandwidth is important information. Thus, the network topology information from OSPF-TE database is collected. The entities in dotted lines are not implemented as part of this thesis.

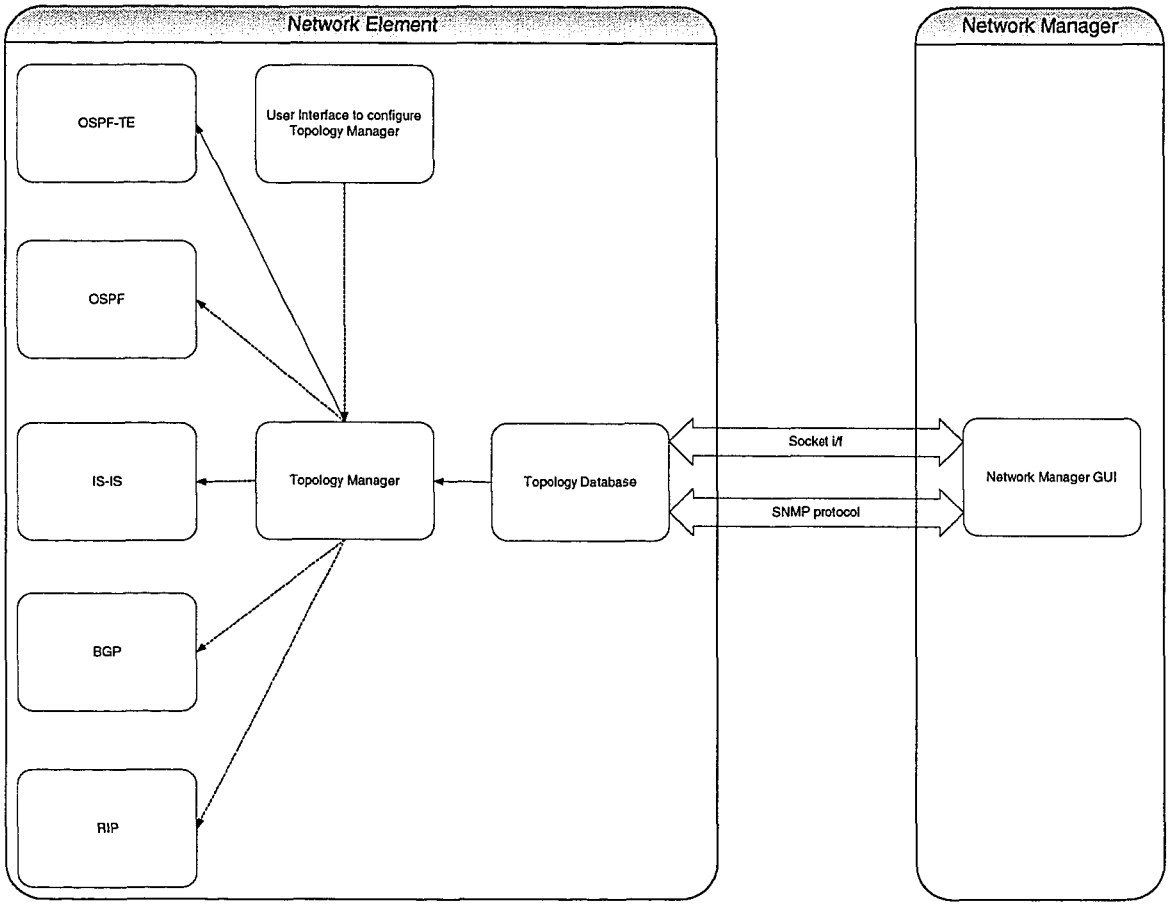


Figure 6: Sources of network topology information

OSPF is a link-state protocol and runs internal to a single Autonomous System (AS) [32]. An autonomous system is a group of routers exchanging routing information using a common routing protocol, called Interior Gateway Protocol (IGP). Each AS has a single IGP. Separate autonomous systems may be running different IGP. OSPF is one of the

first interior gateway protocols to offer load balancing. If a network manager specifies multiple routes to a given destination at the same cost, OSPF distributes traffic over all routes equally. To permit growth, and easily manage the networks, OSPF allow a site to partition its network into subsets called areas. Each area is self-contained and knowledge of the area topology is hidden from other areas. In a multi-access network, there is a designated router that sends link status messages on behalf of the network to the routers attached to the network. Two routers that have interfaces to a common network are neighboring routers. Neighbor relationships are maintained and discovered by OSPF protocol. Adjacency is a relationship formed between selected neighboring routers for exchanging routing information. Link State Advertisements (LSAs) is a record describing the local state of a router or network. LSA of a router includes the state of the router's interfaces and adjacencies. Each link state advertisement is flooded in the routing domain. The collected link state advertisements of all routers and networks form protocol's Link State DataBase (LSDB). The LSAs can be used to derive the network topology information including the link capacity and the available bandwidth.

4.1.4 Network Topology Information Update

Two options available to update the information are push and pull. In the following, merits of each of these approaches in the current context are considered.

In the pull model, the network manager connecting to the router using a TCP/IP socket gathers the latest topology information whenever a request to refresh the network topology is made. As the router is busy with its scheduled workload, this way the router

doesn't have to spend much time on updating the network manager with the latest topology information.

In the push model, in turn, there are two options. Router can send the updates either periodically or as soon as there is a change in topology.

1. Periodic update - the router periodically updates the network manager with the latest topology information. The disadvantage is that there may be no change in the topology but periodically the router spends a considerable amount of time in collecting the information and sending it to the network manager.
2. Update on change - the router sends an update to the network manager whenever there is a change in the topology of the network. The router in addition to its original workload has to spare its time in updating the network manager whenever there is a change in the network topology. If the changes in the topology are too frequent, the router ends up spending significant amount of time in updating the topology changes rather than working on the its core functionality. Although the network manager gets the topology changes immediately, this option results in inefficient use of the router.

To minimize the burden on the router, pull model is opted.

4.2 Displaying Network Topology

The topology information retrieved from the router is of the form as provided by the OSPF-TE link state database [33-34]. The network topology is displayed with nodes and links. For each of the nodes, the corresponding information from the node table in the network manager database is retrieved to display the node with more relevant information

like node name and location. For each of the links, the bi-directional bandwidth of each link is displayed.

Tom Sawyer Layout Toolkit is used to add the graphic layout capabilities to the network topology [35]. The Graphic Layout Toolkit (GLT) acts as a software component. The GLT follows the object-oriented programming methodology and Java standards-based development as such it easily integrates with Java making it more flexible and portable across the platforms. Nodes and edges can be added to the graph. The diagrams display complex relationships in data (nodes and edges) exposing the underlying graph structures, choosing logical positions for nodes and a suitable routing for edges. These diagrams are easily understandable and enhance the visualization applications. GLT is composed of a graph management system, a portable drawing model, and a virtual function-driven layout system. It allows developing a well-designed graph quickly and easily. There is no limit on the topology of the graph as it can contain any number of nodes and edges. GLT uses memory on demand thus reducing the overall memory usage.

4.3 What-if Scenarios

Network administrator would often want to try out what-if scenarios to understand the network behavior. A simple scenario is to compute a non-congested route between a source and a destination, for a given bandwidth. Another scenario is to analyze the traffic on a given route when one or more nodes or links go down and need to find alternative paths. The following aspects were considered.

- Location and scope of computing the path.

- Algorithms to compute the path.

4.3.1 Location and Scope of Computing the Path

First option is enriching the network manager to provide what-if scenarios in a given network without always connecting to the router. In the scenarios, the status of a given node or link can be simulated to be down or inactive. Without directly connecting to the router, the LSP in a what-if scenario is calculated using a routing algorithm. Dijkstra's shortest path algorithm is used when there are no constraints on computing the path. Constraint-based routing algorithm is used when there are constraints in computing the path like minimum bandwidth required on each link of the path. The architecture for this option is described in Figure 7.

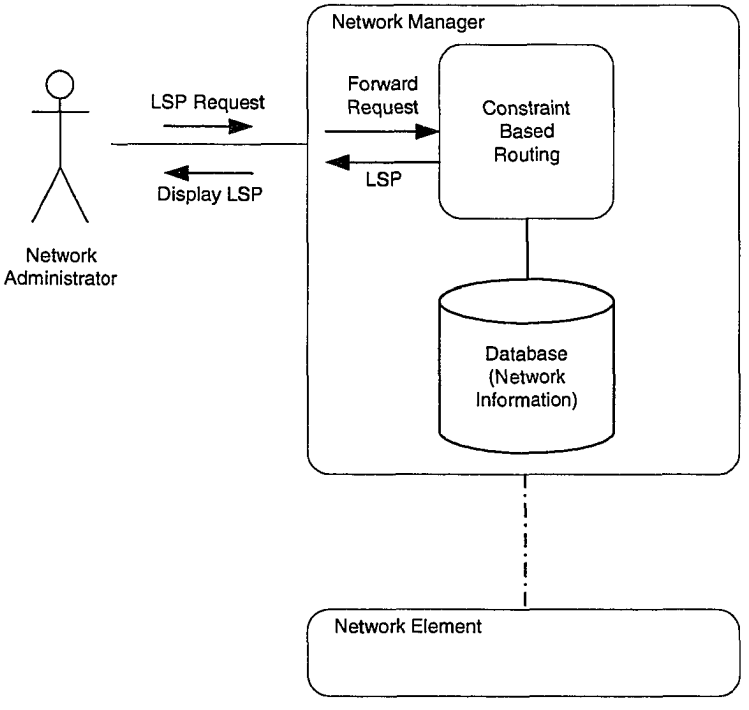


Figure 7: LSP computation without contacting router

Advantage of using this approach is:

a. As the there is no need to connect to a router for determining what-if scenarios, the resources of the router are not wasted.
b. As the network administrator can modify the network topology in the network manager, to be different from that on the router side, different scenarios can be simulated and the best suitable route can be selected. For example, although the network is not currently congested, if the network administrator knows that during peak hours, some of the links can be congested, the network administrator can simulate such what-if scenarios and compute the new paths. Eventually, these new paths can be set on the router to preemptively avoid congestion in the network.

Disadvantages of using this approach are:

a. The network topology may have changed at the router.
b. The algorithm used to compute the path may not be same as that used by the router.
c. The obtained path is not guaranteed in the actual router, as is it not reserved.

Second option is, the network manager sending source and destination to the router when a request for a LSP is made. The request is forwarded to the constraint based routing algorithm Constrained Shortest Path First (CSPF) on the router. The constraint based routing algorithm returns a path based on the OPSF-TE link state Database (LSDB) information. The path is returned to the network manager. However, this path still needs

to be confirmed before it can be set. The network manager can set up the requested path using the CLI. The architecture for this option is described in Figure 8.

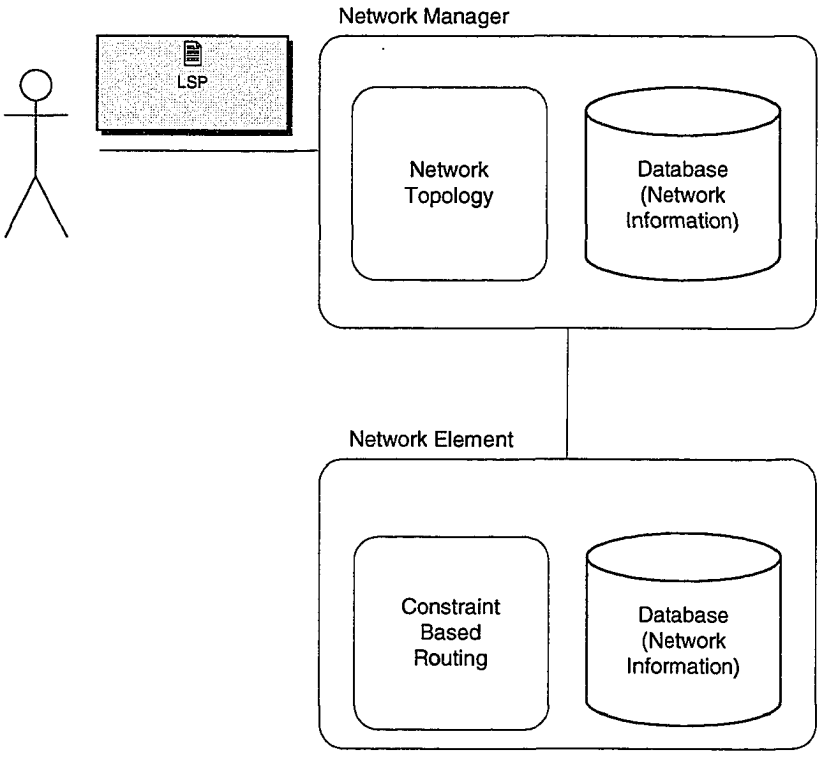


Figure 8: LSP computation at the router, without reserving the path

Advantages of using this approach are:

a. The network topology is the latest as determined by the router.
b. Router can use the same algorithm to compute the what-if scenario as used to compute the path.

Disadvantages of using this approach are:

a. The resources of the router are wasted.
b. The obtained path is not guaranteed in the router, as is it not reserved.

c. The what-if scenarios cannot be simulated with different network topology.

Third option is, when a request for a LSP is made, sending the source and destination to the router. The Connection Manager processes the request on the router. The Constraint Based Routing algorithm calculates the route between source and destination using the OSPF-TE database. The resource manager for each router maintains the up-to-date information about the router like bandwidth on each of the outgoing links. It takes sometime for the updated network topology information about a router to propagate to the OSPF-TE database. Hence, in this option, the Connection Manager first uses Constraint Based Routing algorithm and gets a route between the requested source and destination. Next, Connection Manager contacts the Resource Manager to verify the availability of this route and obtains the confirmed path. Connection Manager then reserves the confirmed path. The confirmed path is returned to the network manager. The architecture for this option is described in Figure 9.

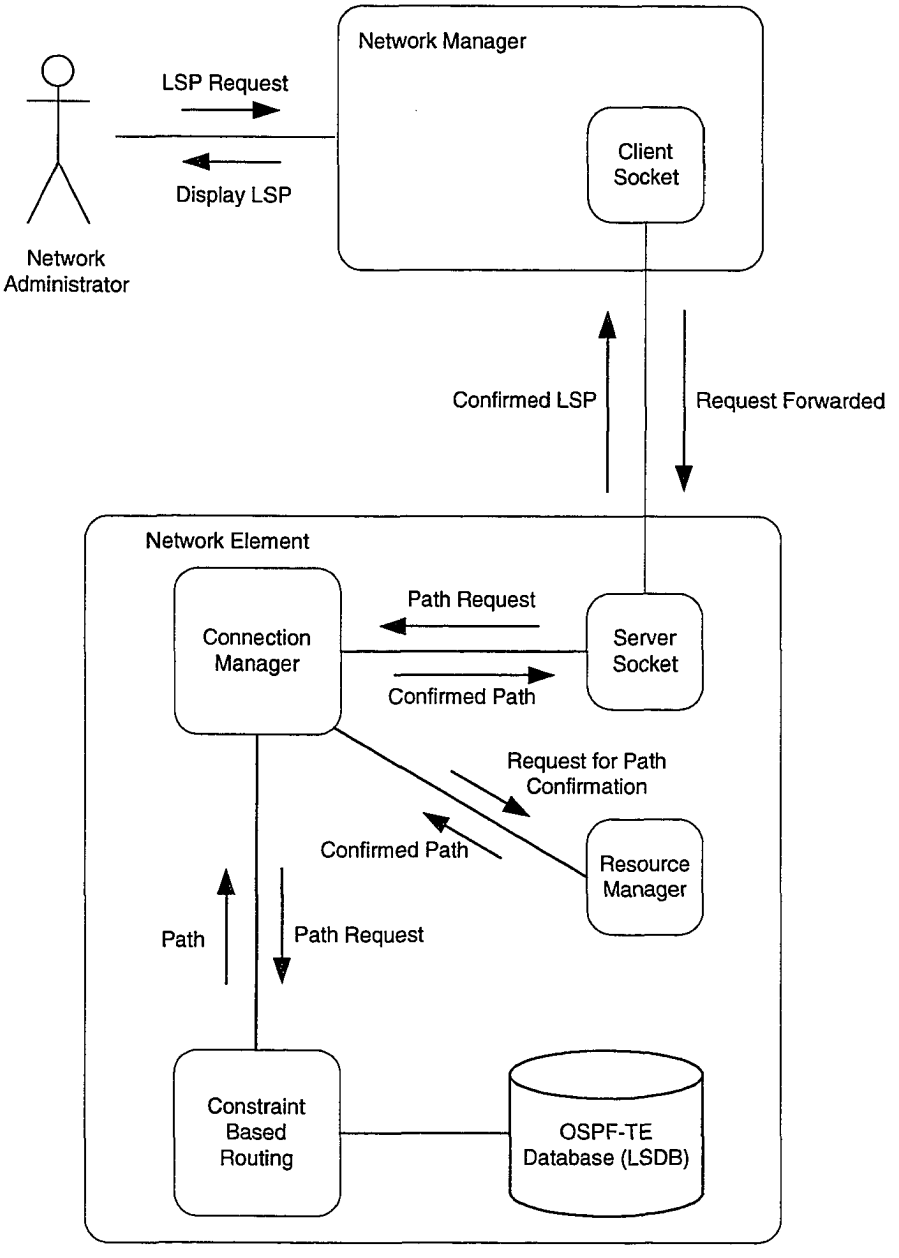


Figure 9: LSP computation at the router, with path reservation

Advantages of using this approach are:

a. The obtained path is guaranteed at the router.

b. The network topology is the latest as determined by the router.

c. Router can use the same algorithm to compute the what-if scenario as used to compute the path.

Disadvantage of using this approach is:

a. The resources of the router are wasted.
b. The what-if scenarios cannot be simulated with different network topology.

It would certainly be good to provide all the options and let the network administrator choose depending upon the situation. As minimizing the unnecessary burden on the router is a major concern, computing the path on the network manager without contacting the router is chosen to be first option and is implemented in this thesis.

4.3.2 Algorithm to Compute the Path

The algorithm used to compute the what-if scenarios are, with specific bandwidth constraint and possible changes in the state of node(s) and/or link(s), to be able to compute a path between given source and destination. This is a form of local constraint-based source routing described in RFC 3630, hereafter called Constraint Based Routing (CBR) algorithm. As suggested in the RFC 3630, MPLS paths are used to instantiate the route. CBR can be NP-hard, depending on the nature of the attributes and constraints.

In this thesis, the following CBR algorithm is used.

1. Isolate the nodes and links that are supposed to be absent.
2. Further prune the network by isolating the links that do not meet the bandwidth constraint.

3. Run Dijkstra's shortest path algorithm on the resulting network to compute the optimum route.

4.4 Command Line Interface

Most of the routers come with an option to configure them using command line interface (CLI). To manage the network element from within the network manager user interface, a telnet port of the network element is opened to execute CLI.

Once the connection is established between the network manager and the network element, the commands to configure the network element are issued in the telnet window of the network manager. These commands are then executed on the network element. The responses from network element are displayed on the network manager telnet window.

CLI is used to set up the LSP after the what-if scenarios are analyzed and a suitable path is chosen. The router then executes RSVP-TE based protocol to establish the path from given source to destination.

When the network administrator selects a path and wants to set it in the network, RSVP-TE is used. RSVP-TE is used to provide quality of service and load balancing across the network. RSVP allows the use of source routing based on CSPF, where the ingress router computes the entire path through the network.

Resource Reservation Setup Protocol (RSVP) is used by a router to request specific QoS from the network for particular application data streams or flows. RSVP is also used by routers to deliver QoS requests to all nodes along the path of the flow and to establish and

maintain state to provide the requested service. RSVP operates on top of IPv4 or IPv6 in the transport layer of the OSI protocol stack. However, RSVP does not transport application data but behaves much like a routing protocol. RSVP itself is not a routing protocol but is designed to operate with other routing protocols. Routing protocols determine where packets get forwarded, whereas RSVP is only concerned with the QoS of those packets that are routed. RSVP with Traffic Engineering extensions (RSVP-TE) is used to establish the paths with constraints. Most of the routers running MPLS traffic use RSVP-TE to establish the LSPs [36-38].

4.5 Web-based Network Management

In a typical remote management scenario, a thin client is installed on a remote node that connects to the management server. Installing this thin client becomes a prerequisite to accessing the management server from anywhere in the network. To be able to invoke the network manager user interface from anywhere in the network, without having to run any kind of thin client, a web server is run on the management server and a browser is used to invoke the network manager interface.

Microsoft's Internet Information Services is used as a web server in this system. The web server hosts the network manager and redirects all the connections to the network manager. The network manager web browser will be launched only for authorized personnel after verifying their user identification (ID) and password, which authenticates them to enter into the network element's network. This feature enables to have web accounts that can be limited based upon department, geographic area or any other criteria. The web-based management gives the advantage that the network manager can be

invoked from anywhere in a connected network, which could very well be outside in relation to the location of the network element.

Chapter 5

Implementation

In this chapter, implementation details of the proposed network manager are presented. Java language is used to program the system on the network manager side using Windows platform. To run Shortest Path First (SPF) algorithm already available in C language, JNI is used. On the router side, the platform is based on Linux operating system. The routing software is primarily based on C/C++ language. The management software modules are developed using C++ and integrated into the routing software.

5.1 Gathering Network Configuration

In the real world, the network topology changes as and when the status of the nodes and links changes. The topology changes also occur as the bandwidth allocation on an outgoing link gets modified based on demand. In order to find a path between a source node and a destination node, the network to which these nodes belong should have latest topology in order to give a genuine path between the two. In the following subsections, details of gathering the network configuration data are provided.

5.1.1 OSPF-TE Information

As described in Chapter 4, section titled Source of Network Topology Information, OSPF TE is chosen as the source of network topology information. In this section, details about OSPF-TE and the transformation of the OSPF-TE information to the data structure used

to display the network topology are provided. In TE extension for OSPF, as defined in [33], database link state advertisement (LSA) for each link consists of the following sub type/length/values (TLVs). The following sub TLVs are defined in OSPF TE Version 2.

1. Link type – the possible link types are point-to-point and multi-access.
2. Link ID – this identifies the other end of the link. For point-to-point links, this is the router ID of the neighbor. For multi-access links, this is the interface address of the designated router.
3. Local interface IP address – IP address(es) of the interface corresponding to the link.
4. Remote interface IP address – IP address(es) of the neighbor's interface corresponding to this link.
5. Traffic engineering metric – network administrator assigned metric for traffic engineering purposes, which may be different from standard OSPF link metric.
6. Maximum bandwidth – in bytes/second, is the link capacity that can be used.
7. Maximum reservable bandwidth – in bytes/second, is the maximum bandwidth that may be reserved on this link in that specific direction. When the link can be oversubscribed, this value may be greater than the maximum bandwidth.
8. Unreserved bandwidth – in bytes/second, is the amount of bandwidth not yet reserved at each of the eight priority levels (0 through 7).
9. Administrative group – a bit mask assigned by the network administrator. Each bit corresponds to one administrator group assigned to the interface. A link may belong to multiple groups.

To display the network topology, router sends the following information to the network manager.

1. Area ID – this field is the area ID of the advertising router, taken from the OSPF routing table.
2. Advertising Router – this field is the router ID of the advertising router, taken from the OSPF routing table.
3. Identifier – this field is the router ID of the neighboring router.
4. Maximum Bandwidth – this field gives the link capacity between the advertising router and the neighboring router in that specific direction (advertising router to neighboring router).
5. Maximum Reservable Bandwidth – this field gives the reservable bandwidth between the advertising router and the neighboring router in that specific direction (advertising router to neighboring router).

5.1.2 Socket-based Management Protocol

The communication between the router and the network element is based on the socket connection. In this subsection, semantics of the management protocol used between the network manager and the router are described.

The network manager connects to the router through a TCP/IP socket to gather the latest topology information whenever a request to refresh the network topology is made. Socket communication between network manager and router creates a channel to send and receive information between the two processes [39]. Stream based socket transport

mechanism (TCP) is used as it provides reliability of the communication using retransmission of any lost packets. TCP is preferred compared to datagram transport mechanism (UDP), which does not guarantee reliable communication. Using connection-oriented transport, network manager initiates the connection and acts as a client process. The router, on the other hand, receives the connection and acts as a server. The server process on the router creates a socket, maps to the router IP address, and waits for client requests. The client process on the network manager creates its own socket and determines the location specific information (such as host IP address and port number) of the server. In each request, client sends the type of request to the server. Several types of requests are defined.

- Get network topology
- Get LSP from CSPF
- Get LSP from Connection Manager

Although the APIs were provided for all types of requests, router currently honors only network topology requests. The router gathers the current topology information from the link state database of the OSPF-TE protocol as described in this Chapter, section titled OSPF-TE Information. Each field of the LSA record is separated by a space, and each LSA record is separated by a new line in the string buffer response. The server socket after gathering the information passes on the same to the client socket of the network manager.

5.2 Displaying Network Topology

In this section, using the information gathered from the router, the approach used to display the network topology at the network manager is presented. To display the network topology, the network manager uses a database to store network information as described in the next subsection.

5.2.1 Network Manager Database

Network manager stores static data about the network in Microsoft Access Database. The database has the following tables.

1. Node table – this table contains all the information relevant to the Node, shown in Table 2. Tables related to Node table are Node Type table and Network State table.

#	Column Name	Details
1	Network ID	Corresponds to the area ID of the OSPF-TE protocol.
2	Node ID	The ID of a node in a particular network.
3	Node Type ID	The type of node, router or switch
4	Node Name	The name of the node.
5	IP Address	Maps to advertising router ID of the OSPF-TE protocol.
6	Node Province	The province in which the node is physically located.
7	Node Country	The country in which the node is physically located
8	Node Latitude	The latitude of the place where the node is physically located
9	Node Longitude	The longitude of the place where the node is physically located
10	Node state ID	The status ID of the Node (which maps to active, inactive or congested)
11	Last Updated	The date when the node table was last updated
12	Updated By	The network administrator who last updated this table.

Table 2: Node table

2. Edge table – this table has information relevant to the link. Two tables relevant to Edge table are Edge Type table and Network State table.

#	Column Name	Details
1	Network ID	Corresponds to the area ID of the OSPF-TE protocol.
2	Edge ID	The ID of an edge in a particular network.
3	Edge Type ID	The type of edge, ATM or OC3 or Frame Relay or Ethernet.
4	Edge Name	The name of the edge.
5	Edge From Node ID	Between every two nodes that have a link, a bi-directional edge is created. This field corresponds to the From Node ID of the edge (i.e., the ID of the node from where the edge is starting).
6	Edge To Node ID	This field corresponds to the From Node ID of the edge (i.e., the ID of the node from where the edge is ending).
7	Edge state ID	The status ID of the edge (which maps to active, inactive or congested).
8	Edge Max Bandwidth	The maximum bandwidth on the edge (from From Node ID to To Node ID).
9	Edge Avl Bandwidth	The bandwidth available on the edge (from From Node ID to To Node ID).
10	Edge Cost Metrics	The cost of an edge in a route is described by a single dimensionless metric.
11	Reverse Edge Max Bandwidth	The maximum bandwidth of the edge in the reverse direction (from To Node ID to From Node ID).
12	Reverse Edge Avl Bandwidth	The bandwidth available on the edge in the reverse direction (from To Node ID to From Node ID).
13	Reverse Edge Cost Metrics	The cost of an edge in a route is described by a single dimensionless metric in the reverse direction.
14	Last Updated	The date when the edge table was last updated
15	Updated By	The network administrator who last updated this table.

Table 3: Edge table

3. Node Type table – this table has information relevant to the node and maps node type ID to node type name.

#	Column Name	Details
1	Node Type ID	Node Type ID, an integer
2	Node Type Name	Name of the node type (such as Router or Switch)
3	Last Updated	The date when the Node Type table was last updated
4	Updated By	The network administrator who last updated this table.

Table 4: Node Type table

4. Edge Type table – this table has information relevant to the link and maps link type ID to link type name.

#	Column Name	Details
1	Edge Type ID	Edge Type ID, an integer
2	Edge Type Name	Name of the edge type (such as Ethernet, OC3, Frame Relay)
3	Edge Type Bandwidth	Default bandwidth for the edge type (such as 10MB for Ethernet)
4	Last Updated	The date when the Node Type table was last updated
5	Updated By	The network administrator who last updated this table

Table 5: Edge Type table

5. Network State table – this table has information relevant to the state of the network and maps network state ID to network state name.

#	Column Name	Details
1	Network State ID	State ID, an integer
2	Network State Name	Name of the network state (Active, Inactive and Congested)

Table 6: Network State table

5.2.2 Accessing Network Manager Database

After gathering the latest topology information from the router, the network manager interprets the data so that it can display the network topology of that particular network. In preparation for the displaying the topology information, a few intermediate files are created. The following paragraphs explain the operation.

1. Node/Edge Status text file – contains the records for node and edge status.
    a. Each Node Status record has the following fields: <network ID, node name, node type, and node status>.

b. For each entry of the LSA, using the area ID and the advertising router ID, fetch the corresponding record in Node table of network manager database.

c. The node name and the node type are collected from the database and written to Node/Edge Status text file along with network (or area) ID.

d. Since the router flooded the network with this LSA, status of the advertising router ID is set to active.

e. Each Edge Status record has the following fields: <edge ID, edge type, and edge status>.

f. For each entry of the LSA, using the area ID, advertising router ID, and neighbor ID, fetch the corresponding record in Edge table of network manager database.

g. The edge ID and the edge type are collected from the database and written to Node/Edge Status text file.

h. Since the advertising router sent out this LSA, status of the edge ID is set to active.

The text file thus contains the latest status information received from LSA for each node and edge, as shown in Figure 10.

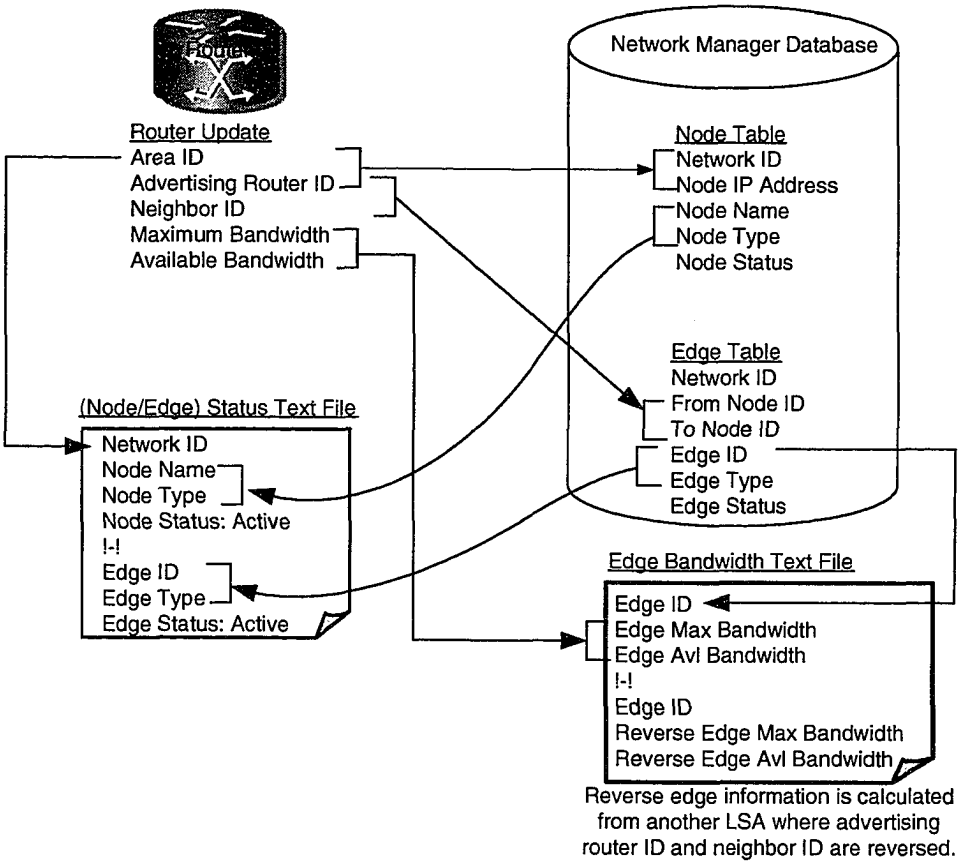


Figure 10: Node/Edge Status, and Edge bandwidth computation

2. Edge Bandwidth text file

   a. Each Edge Bandwidth record has the following fields: <edge ID, edge maximum bandwidth, and edge available bandwidth>.

   b. For each entry of the LSA, using the area ID, advertising router ID, and neighbor ID, fetches the corresponding record in Edge table of network manager database.

   c. The edge ID is collected from the database and written to Edge Bandwidth text file. Edge maximum bandwidth and edge available bandwidth from the LSA are also written to the Edge Bandwidth text file.

d. Similar record with for the reverse edge is computed from another LSA where advertising router ID and neighbor router ID are reversed and written to the same file.

The text file thus contains the latest bandwidth information as received from LSAs for each edge in both directions, as shown in Figure 10.

Using the data collected in intermediate files, the Node and Edge data structures of the network manager are updated. If there are any edges or nodes in the network manager database but there are no corresponding LSAs, status of such nodes and links are set to inactive.

5.2.3 Displaying Network Topology

The information about the nodes and links gathered from the router are compared to the nodes and links already in the network manager's database. The nodes and links that are both in router's update as well as in the network manager's database are respectively updated with status information as well as bandwidth information. The nodes and links that are in network manager's database but not in the router's update are shown as inactive. The nodes and links that are in router's update but not in network manager's database are considered as new nodes and links that have come into existence after the network manager's last database update. In this work, it is assumed that the network manager updates its database with the information on new nodes and edges before they can be shown in the network graph. Although it is possible to display new nodes and links that are not yet available in the network manager's database, the display would only

show that there are new nodes and links without much meaningful information like node name and other properties. In the implementation, if such nodes or edges appear in LSAs, they are ignored. Table 7 captures the node or link display approach. The maximum and the available bandwidth on each link are shown in both directions.

Using Tom Sawyer GLT, the network topology is visually displayed. The network manager database is updated with the new network information.

#	**Router update information**	**Network manager database**	**Network topology display approach**
1	Node/link does not exist in LSA	Node/link exists in database	Display the node/link as *Inactive*
2	Node/link exists in LSA	Node/link does not exist in database	*Ignore* the node/link as there is no information to display.
3	Node/link exists in LSA	Node/link exists in database	Display the node/link as *Active*

Table 7: Node or link display approach

5.3 What-if Scenarios

What-if scenario analysis is very powerful tool for network administrators to understand the network behavior that may arise in future. In the following subsections, collecting different what-if scenarios from the network administrator, computing the paths based on Constraint Based Algorithm, and setting up of the paths are discussed.

5.3.1 User Interface to get What-if Scenarios

There are many possible what-if scenarios that network administrator may like to carryout. In this subsection, some of the most useful scenarios are considered.

1. Zero or more node failures.
2. Zero or more link failures.
3. Minimum bandwidth requirement can be zero or more.

In each of these combinations the following path requests may be made.

1. Edge redundant path
2. Node redundant path

For example, a node may be selected to be down. And, using the user interface, specify a source and a destination, the minimum bandwidth on the path, and request a node redundant path.

5.3.2 Constraint Shortest Path First Algorithm

In an autonomous system, which is a collection of nodes in a network, the routes are determined by intra-domain routing protocols such as OSPF and RIP. In an intra-domain routing protocol, the path computation is based on an algorithm that optimizes a particular scalar metric. In case of OSPF, this metric is the administrative metric of a path. The network administrator assigns an administrative metric to each link in the network.

In addition to finding a path that is optimal with reference to some metric, often there is a requirement to satisfy some of other constraints. Constraint Shortest Path First (CSPF) routing algorithm is used in conjunction with OSPF to find an optimal path that meets a given set of constraints along the path [1]. In this thesis, a routing path is computed

between a given source and a destination with minimum bandwidth requirement on each link of the path. When no bandwidth requirement is specified, any optimal path is acceptable. When a non-zero minimum bandwidth requirement is mentioned, each link along the path should meet this constraint. Other constraints that may be added are failure of one or more nodes and/or links, just to try out what-if scenarios and get other possible paths if available.

In a network, when there is a choice of multiple paths to a given destination, Dijkstra's algorithm is used. Dijkstra's algorithm computes a path between a source and a destination such that the administrative metric of the path is minimized. Dijkstra's algorithm is described below.

1. Fix the source node
2. Define a set S of nodes, and initialize it to empty set. As the algorithm progresses, the set S will store those nodes to which a shortest path has been found.
3. Label the source node with 0, and insert it into S.
4. Consider each node not in S connected by an edge from the newly inserted node. Label the node not in S with the label of the newly inserted node + the length of the edge. If the node not in S was already labeled, its new label will be minimum (label of newly inserted node + length of edge, old label)
5. Pick a node not in S with the smallest label, and add it to S.
6. Repeat from step 4, until the destination node is in S or there are no labeled nodes not in S.

In this thesis, Constraint Based Routing uses the following CSPF algorithm.

1. Initially, a network link weight matrix is created. For a network with N nodes, the matrix size is NxN.
2. In this matrix, if there is a link between a pair of nodes, the cost or metric of the link is set as the link weight. If there is no link between two nodes, the link weight is set to INFINITY (a very high value).
3. A copy of the original link weight matrix is created and rests of the steps are carried out.
4. If a constraint is node failure, all the links corresponding to the node are set to heavy weight (a high value).
5. If a constraint is link failure, the link weight is set to a corresponding to the node are set to heavy weight (a high value).
6. If there is a constraint, such as bandwidth, prune the link weight matrix to eliminate the links that do not qualify the constraint.
7. Pass the copy of link weight matrix to Dijkstra's shortest path algorithm to find primary node- or edge- redundant path.
8. Set the link weight corresponding to the edges involved in the primary node- or edge- redundant path to heavy weight (a high value).
9. Pass the copy of link weight matrix to Dijkstra's shortest path algorithm to find alternate node- or edge- redundant path.

Note that bandwidth constraint is treated as a soft constraint (if not possible to find a path, a path without meeting the constraint will be displayed) whereas node or link failures are treated as hard constraints (it is not possible to include failed node or link in the path).

5.4 Command Line Interface

Most of the routers usually allow a mechanism to open a port and configure itself. To configure the router right from the network manager user interface, a feature is provided to open the router CLI.

From the network topology displayed, any router can be chosen and a request can be made to invoke the CLI on that router. The IP address and port number of the router CLI are used to invoke the CLI command using *exec* method of Java Runtime library.

In the context of the thesis, setting up of LSP uses the CLI established with the router. Executing a sequence of CLI commands result in setting up of a LSP in a MPLS network.

5.5 Web-based Network Management

To give the option of invoking network manager user interface from anywhere in the network manager's network, web-based network management is provided.

As the network manager is running on Windows platform, Microsoft's Internet Information Services (IIS) is used as a web server in this system. The network manager software is bundled with IIS to invoke user interface from a designated port on the host running the network manager.

The web browser allows authorized personnel to launch network manager after verifying their user identification (ID) and password. When a connection is established to the designed web port of host the network manager is running, the web server redirects all the connections to the network manager.

5.6 Complexity Analysis

In this section, the complexity of the algorithms used in this work is discussed. The algorithms are categorized into communication between router and the network manager, front-end, and back-end.

5.6.1 Gathering Network Configuration

When topology information has to be displayed, the network manager sends the network identifier on the socket connection to the router. The socket module on the router gathers the network topology from router LSDB. The data transferred between the router and the network manager is analyzed in this section. In LSDB, the network topology is stored in terms of connected edges at each node. For n nodes in the network, there are n entries in the LSDB. At each node, there will be information about a set of edges incident to the node. The total of all edges in the LSDB reflect $2m$ bi-directional edges information where m is the number of edges in the network.

5.6.2 Displaying Network Topology

To display the network topology, node/edge status text file and edge bandwidth text files are constructed. Using the topology gathered from the router, network database is queried to collect this information. Node/edge status text file involves querying the node information for each of the n nodes and each of m edges in both directions, resulting in a total of $(n+2m)$ queries. Edge bandwidth text file involves querying each of m edges in both directions, resulting in a total of $2m$ queries. The information in these temporary

files is then transformed to the corresponding node and edge data structures involving n and $2m$ updates respectively.

Displaying the network topology using Tom Sawyer GLT software involves painting the nodes and then the edges with complexities of n and m respectively.

5.6.3 What-if Scenarios

The other important contribution of this work is what-if scenario analysis. In this, user specifies the constraints that need to be satisfied while requesting a node or edge redundant path. Constraint Based Routing algorithm in conjunction with Dijkstra's is used. There are several steps involved in pruning the network link weight matrix. The cost involved in eliminating the node is at most $O(n)$ and that of a link is $O(1)$. In this work, Dijkstra's algorithm is used as an API from JNI, pruning is carried out first and then Dijkstra's algorithm is invoked. Pruning the edges that meet the constraint such as bandwidth is $O(n^2)$ as it involves going through all the links of the network link weight matrix and setting certain links to heavy weight. As it is implemented now, Dijkstra's algorithm has the complexity of $O(n^2+m)$ [40]. Thus the overall complexity of the constraint based routing algorithm is same as that of Dijkstra's.

However, as explained in [40], for sparse graph with number of edges much less than n^2, by implementing minimum priority queue with Fibonacci heap, complexity of Dijkstra's algorithm can be improved to $O(n\log n+m)$. In such a case, instead of performing pruning the network link weight matrix before passing it to the Dijkstra's algorithm, the constraint

can be validated when each edge is added to the path in Dijkstra's algorithm. This will keep the overall algorithm complexity to be same as that of Dijkstra's.

5.7 Results

In this section, step-by-step results of using the integrated network management system developed along with screen shots and an example what-if scenario is presented.

5.7.1 Displaying Network Information

When the network management system is launched, the administrator has an option to select the network. The geographical view of a network available in the database is displayed as shown in Figure 11.

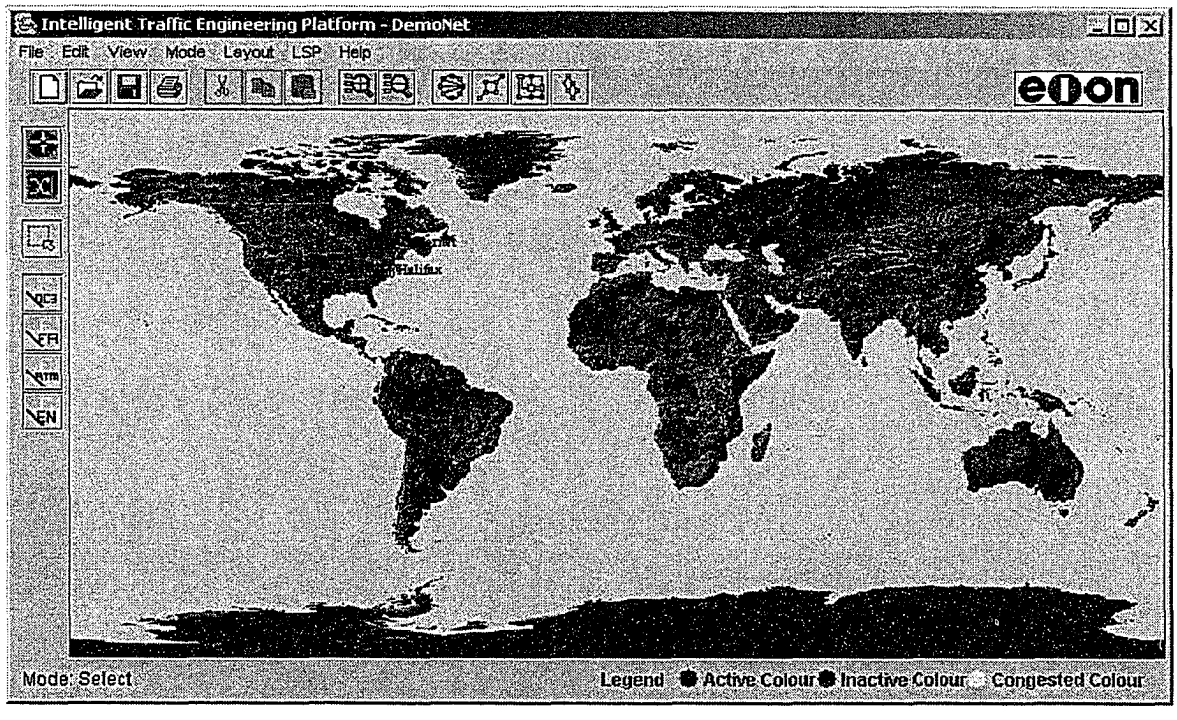


Figure 11: Geographical view of a network

Topological view of the network can be obtained for the same network as shown in Figure 12.

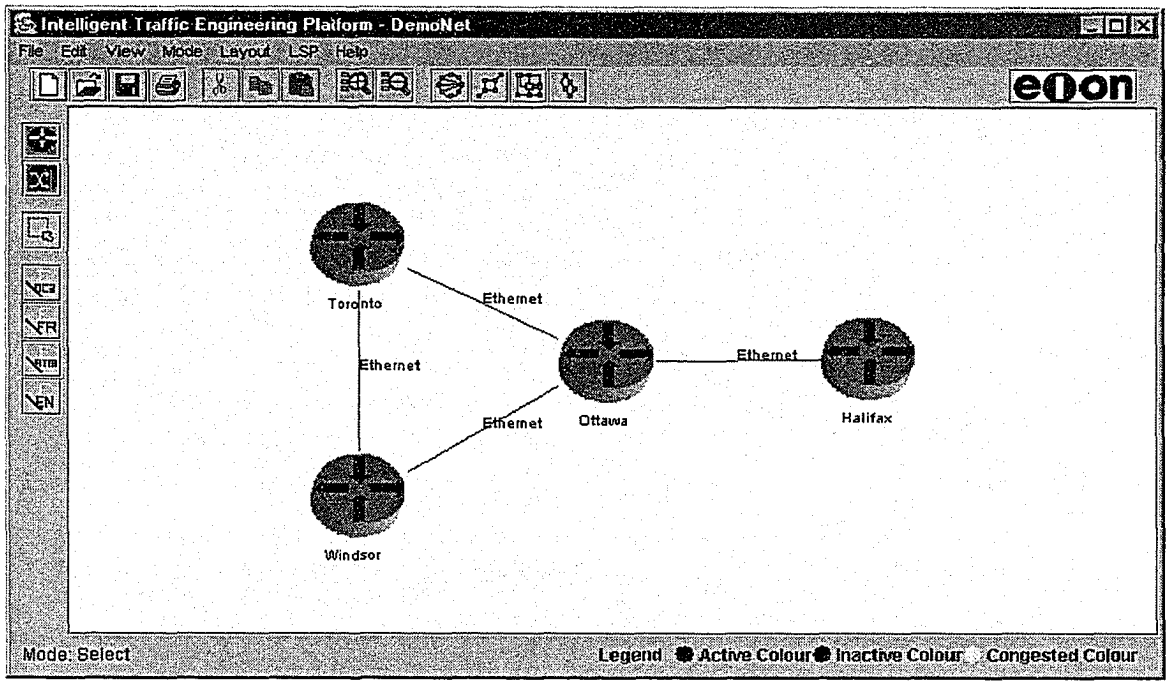
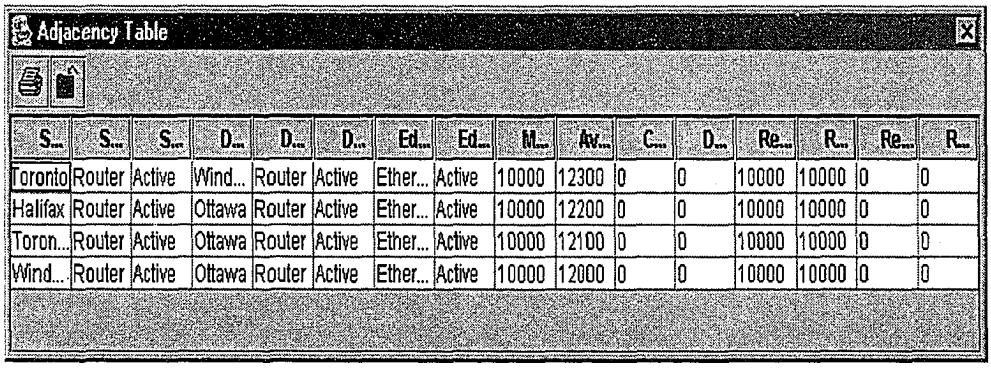


Figure 12: Network topology of the network

Adjacency matrix for the same the network is represented as shown in Figure 13.



S...	S...	S...	D...	D...	D...	Ed...	Ed...	M...	Av...	C...	D...	Re...	R...	Re...	R...
Toronto	Router	Active	Wind...	Router	Active	Ether...	Active	10000	12300	0	0	10000	10000	0	0
Halifax	Router	Active	Ottawa	Router	Active	Ether...	Active	10000	12200	0	0	10000	10000	0	0
Toron...	Router	Active	Ottawa	Router	Active	Ether...	Active	10000	12100	0	0	10000	10000	0	0
Wind...	Router	Active	Ottawa	Router	Active	Ether...	Active	10000	12000	0	0	10000	10000	0	0

Figure 13: Adjacency matrix of the network

Total and available bandwidth information on each direction of the edge can be displayed as shown in Figure 14.

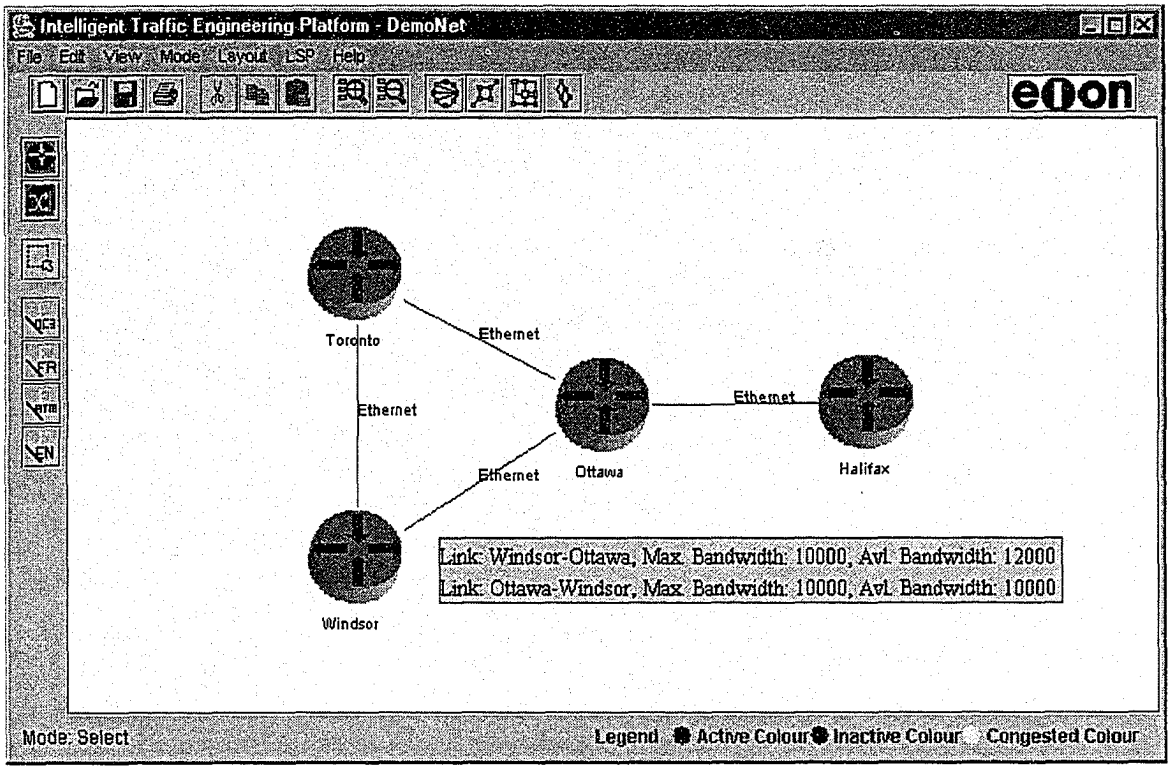


Figure 14: Bandwidth information of a link in both directions

5.7.2 Refreshing Network Topology Information

When network administrator wants to refresh the network topology, area id of the network is sent to the router.

Router gathers network topology information from LSDB, which comprises of LSA's advertised from the routers in the network. A sample LSA entry for a given OSPF area is:

LS age: 6
Options: TOS66
LS Type: Opaque Area Local Links
Link State ID: 1.0.8.6
Advertising Router: 192.168.21.100
LS Seq Number: 80000004
Checksum: 0x8BBA
Length: 124
Link Type: PointToPoint
ID: 192.168.8.5
Local If: 0.0.0.0
Remote If: 0.0.0.0
Te Metric: 0
Max Bw: 10000
Max Res Bw: 2000
Class: 0
Unres Bw: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0

The network topology information sent from router to network manager is in the form of area id followed by multiple rows of advertised link data:

<area id>
<advertising router> <identifier> <maximum bandwidth> <reservable bandwidth>

A part of data transferred from router to network manager is:

2
192.168.31.100 192.168.8.5 10000 2000
192.168.8.5 192.168.31.100 10000 3000

On receiving the current network topology data from router, the network manager analyzes and compares with the existing topology in the database. The refreshed topology appears as shown in Figure 15.

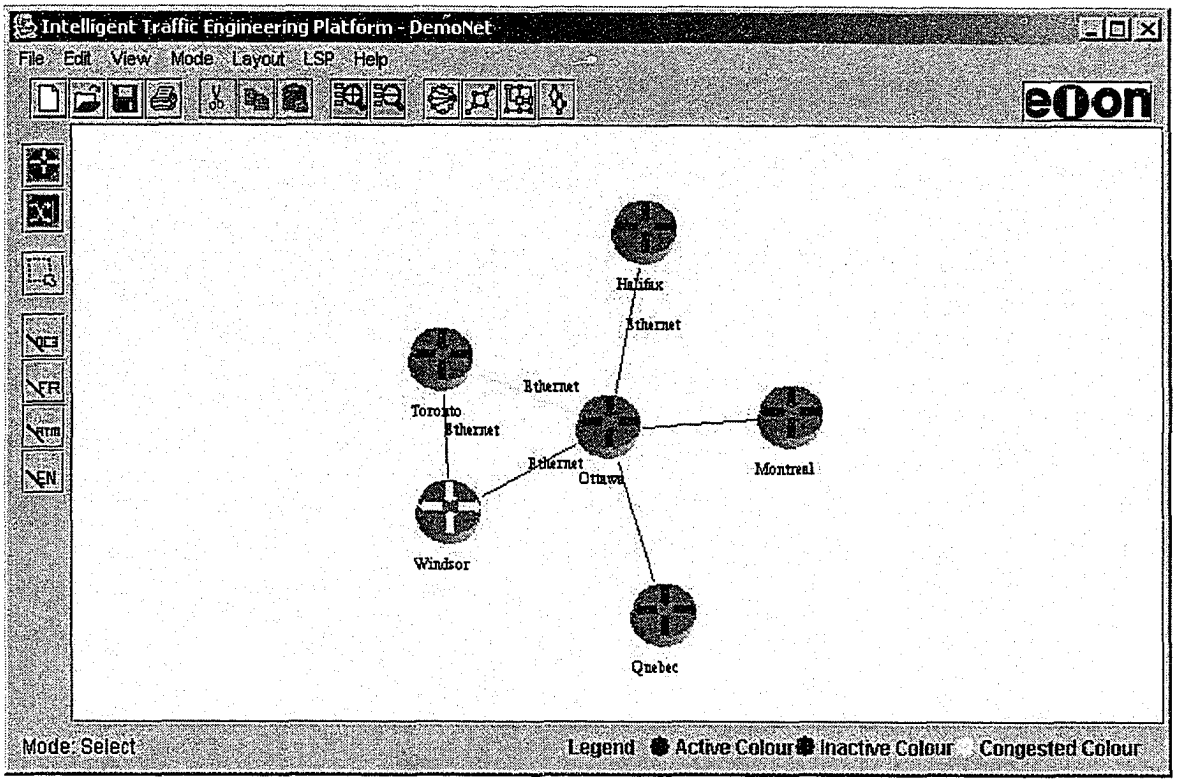


Figure 15: Refreshed network topology of the network

5.7.3 What-if Scenarios

In this section, what-if scenarios are described. The network shown in Figure 16 is used.

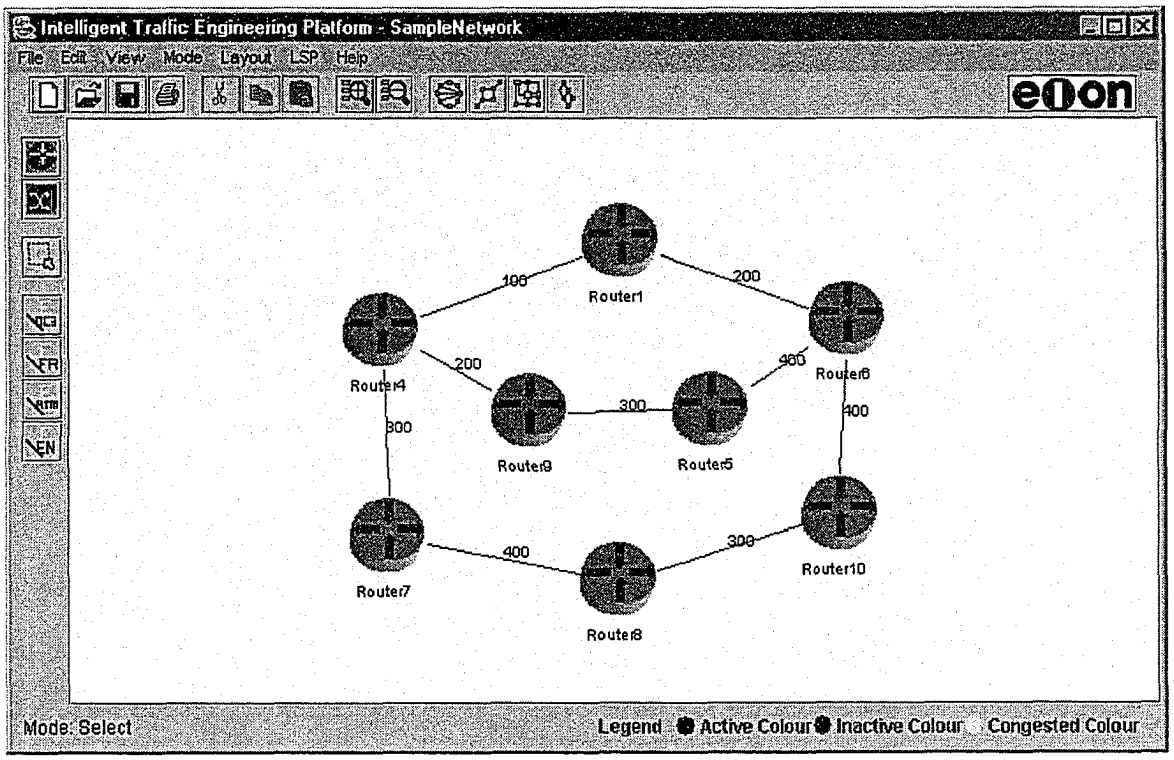


Figure 16: Sample network to analyze what-if scenarios

As shown in Figure 17, at any time, network administrator can select a node or a link and set to active or inactive. This allows analyzing the network behavior under different circumstances without physically bringing down the network elements.

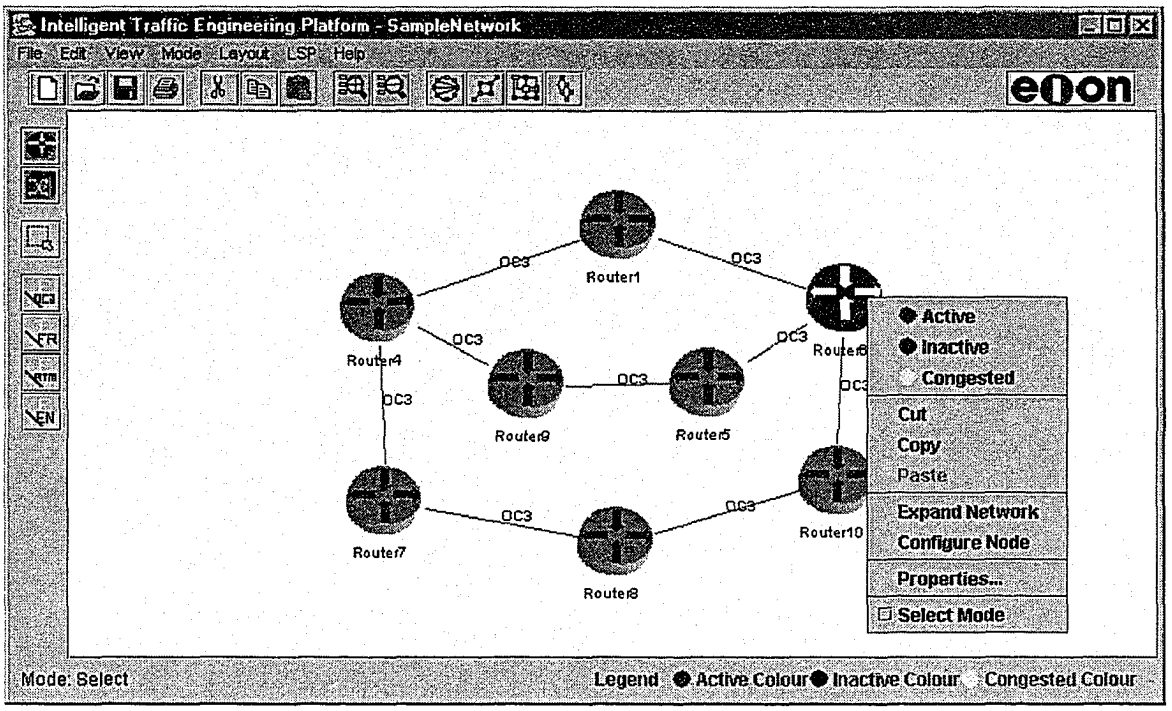


Figure 17: What-if scenario – setting any node/link active/inactive

In addition, a user interface is used to specify the constraints in generating the primary and alternative edge- or node- redundant paths. Two runs are shown – one with zero and another with non-zero bandwidth requirement.

Figure 18 shows the user interface with zero bandwidth constraint.

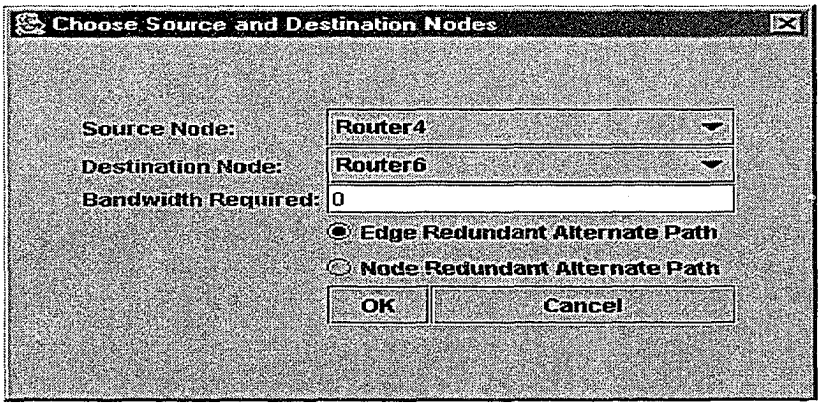


Figure 18: What-if scenario – user interface with zero bandwidth constraint

Using the Constrained Based Routing algorithm, the primary and alternate edge redundant paths are shown in Figure 19.

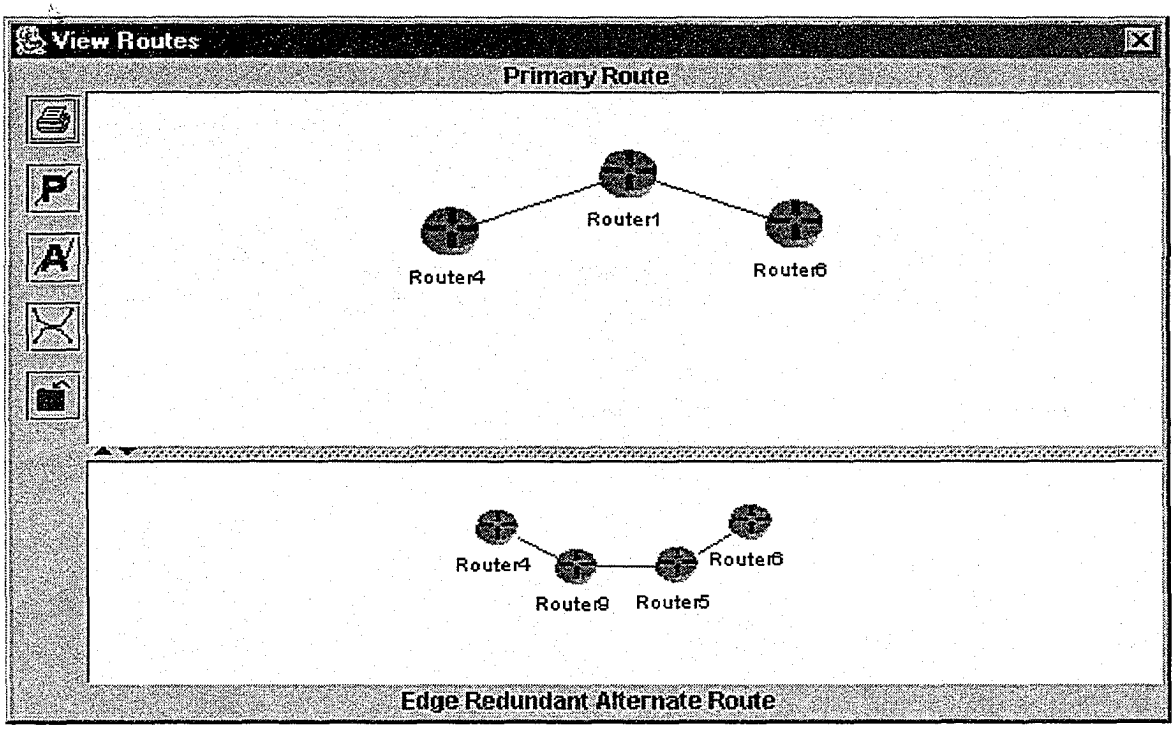


Figure 19: What-if scenario – primary and alternative paths with no constraints

Figure 20 shows the user interface with non-zero bandwidth constraint.

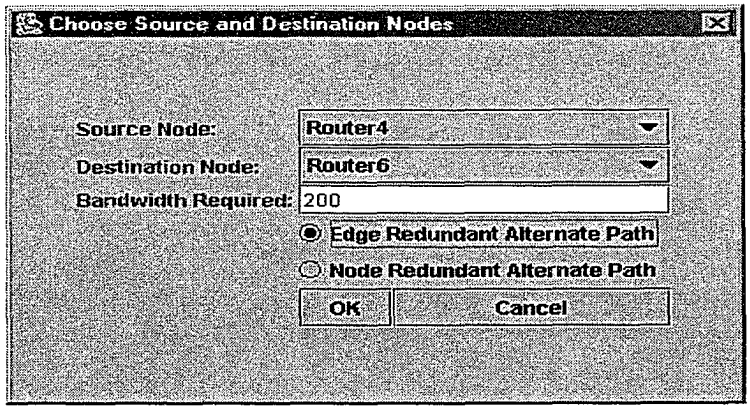


Figure 20: What-if scenario – user interface with non-zero bandwidth constraint

Using the Constrained Based Routing algorithm, the primary and alternate edge redundant paths are shown in Figure 21.

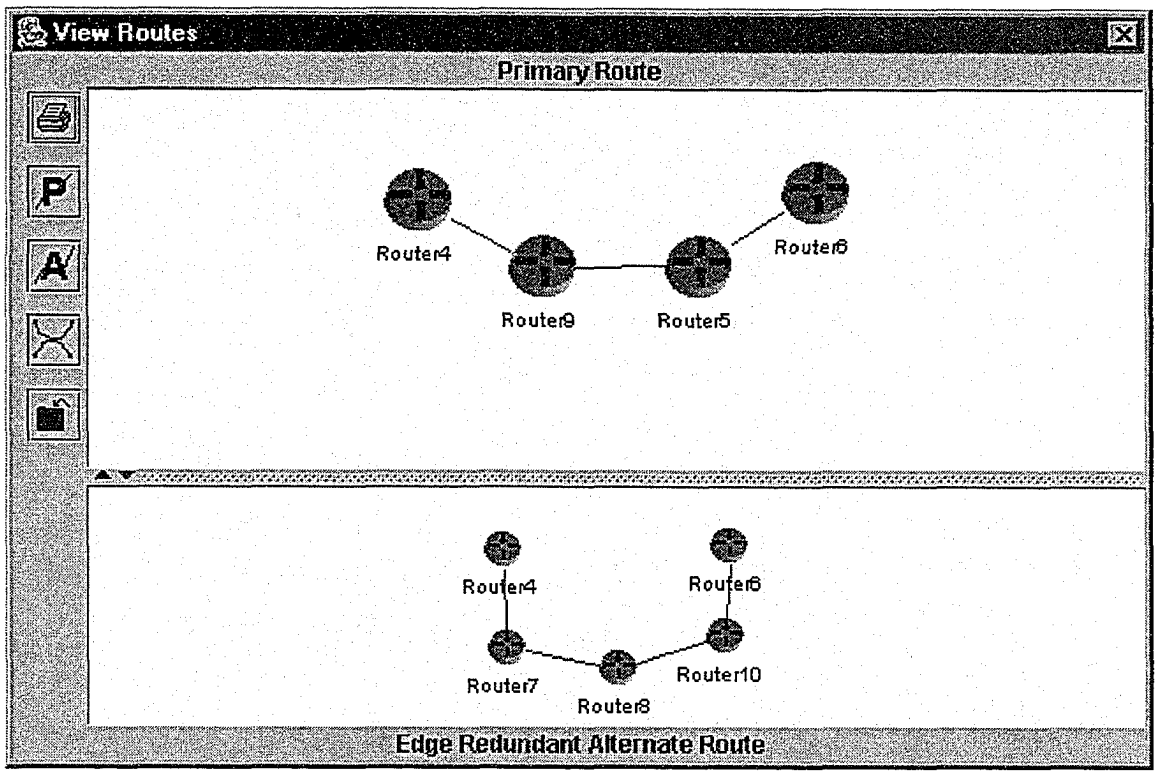


Figure 21: What-if scenario – primary and alternative paths with bandwidth constraint

5.7.4 Command Line Interface

At anytime, to configure a router, network administrator can right click the router icon on the network topology and select "Configure Node" option as shown in Figure 22.

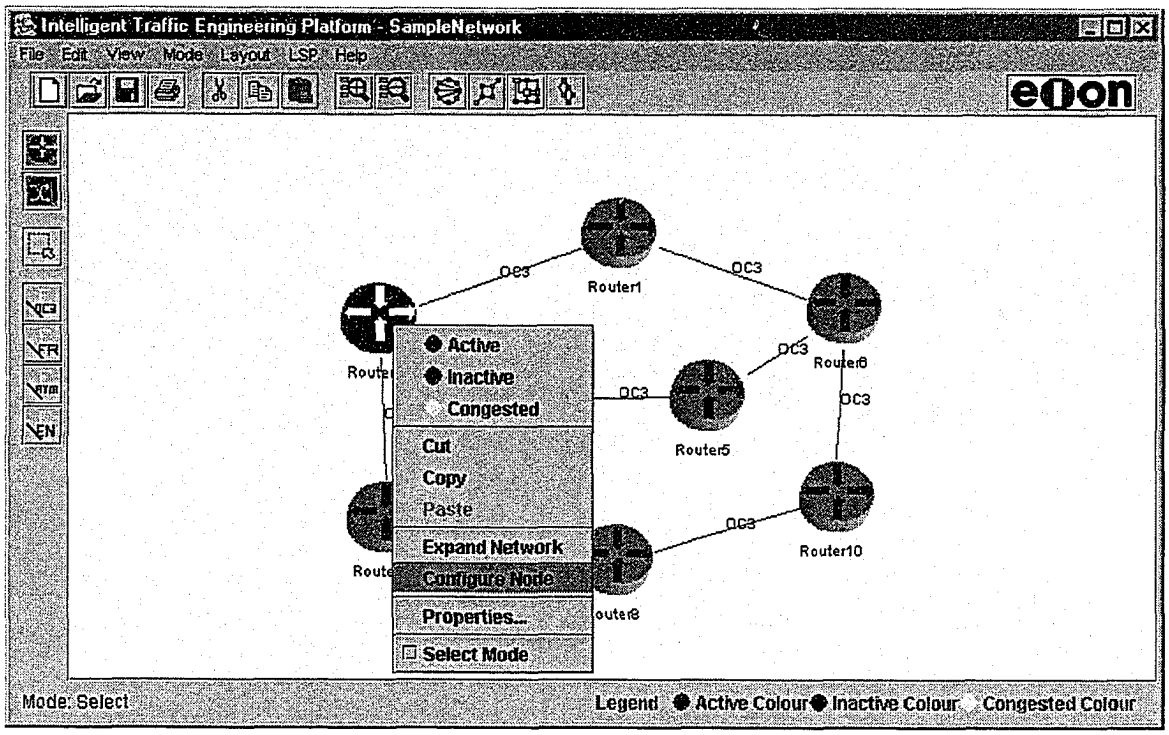


Figure 22: Router Command Line Interface invocation

Selecting this option connects to the router CLI.

Figure 23 captures the command line interface used to configure the Router.

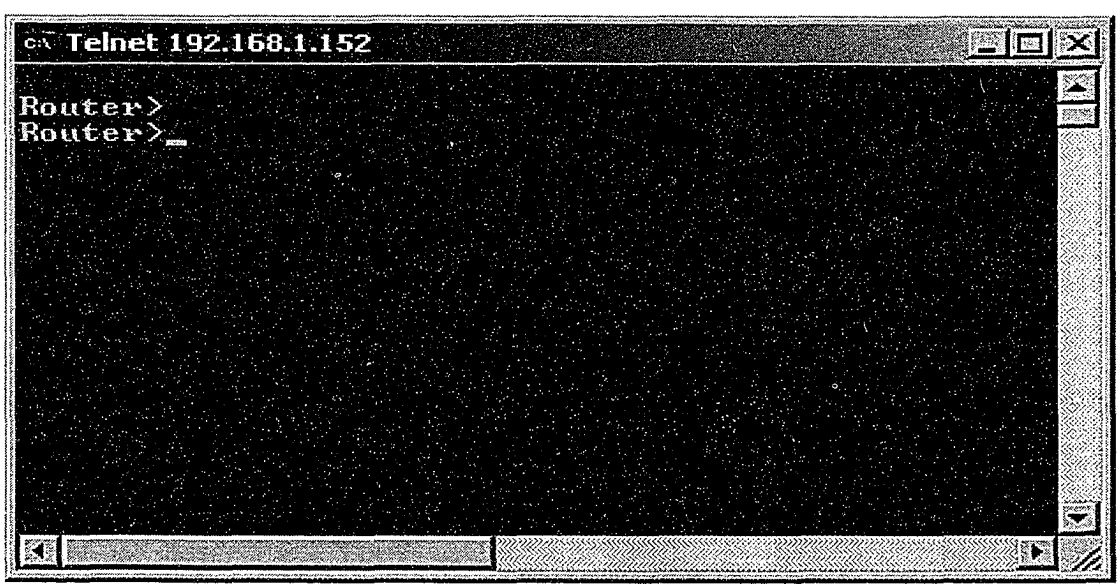


Figure 23: Router Command Line Interface

On the router CLI, executing the commands in sequence as shown in Figure 24 sets up LSP in an MPLS network.

```
cm tunnel id 1
cm tunnel lsp 1
cm tunnel dest 192.168.10.10
cm tunnel send 12.12.30.30
cm tunnel traffic-specific 80 100
cm tunnel priority 4 4 1
```

Figure 24: Router CLI commands to set MPLS LSP

5.7.5 Web-based Interface

From any host in the network, a web-browser can be opened and pointed to the server running the integrated network management system. Figure 25 captures the invoked web-based user interface.

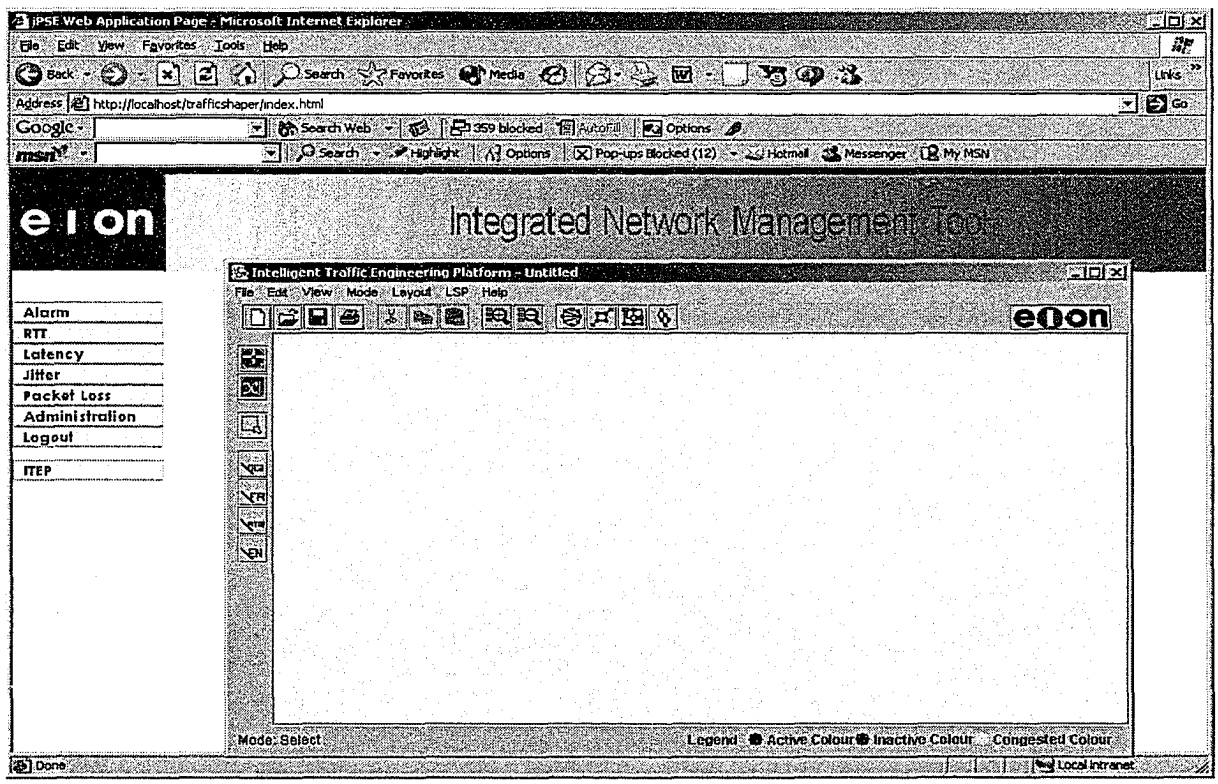


Figure 25: Accessing integrated network management system using web interface

Chapter 6

Performance Analysis

We analyze the performance of network manager, router and the communication between them. End-to-end elapsed time is an important metric since it provides the information on the time taken to setup a path from the time network manager gives the command. The total time taken to refresh a network (end-to-end elapsed time) in the network manager with the latest topology information is the sum of the following five components:

- Time from when the request is made in the network manager to the time it sends the request to the router.
- Time taken for the request from the network manager to be communicated to the router.
- Time taken by the router to gather the latest topology information from the protocol database (LSDB) of the routing protocol OSPF-TE.
- Time taken for the communication of the latest topology information between router and the network manager.
- Time taken by the network manager to analyze the received data and display the refreshed network.

The performance factors of importance here are:

- Amount of time spent by the router in management activity (which should be minimum)

- Management data transferred between router and network manager (which should be minimum)
- The refresh rate of network configuration at the network manager (which should be minimum)
- Amount of time consumed by CBR algorithm in excess of Dijkstra's shortest path algorithm (which should be minimum).

In Section 6.1, data transfer between router and network manager is analyzed. Section 6.2 analyzes the end-end performance in refreshing the network topology. Section 6.3 provides the analysis of the Dijkstra's algorithm and the CSPF used in the path computations between a given source and destination. Section 6.4 describes detailed measurement results.

6.1 Data Transfer

In the first scenario, the item chosen for performance analysis is management data transferred as a function of number of edges for different values of hello message intervals. This scenario is applicable in option when management data is transferred from router to the network manager on change in network topology. Note that, the amount of topology information depends on the number of edges and not on the number of nodes because LSDB stores router interface information.

Total amount of management data sent in bytes/sec = N * f * e * m, where

N= number of bytes of management data per field (assumed 4 because IP address is 4 bytes),

f = number of data fields collected per edge,

e = number of edges, and

m = number of hello messages/sec.

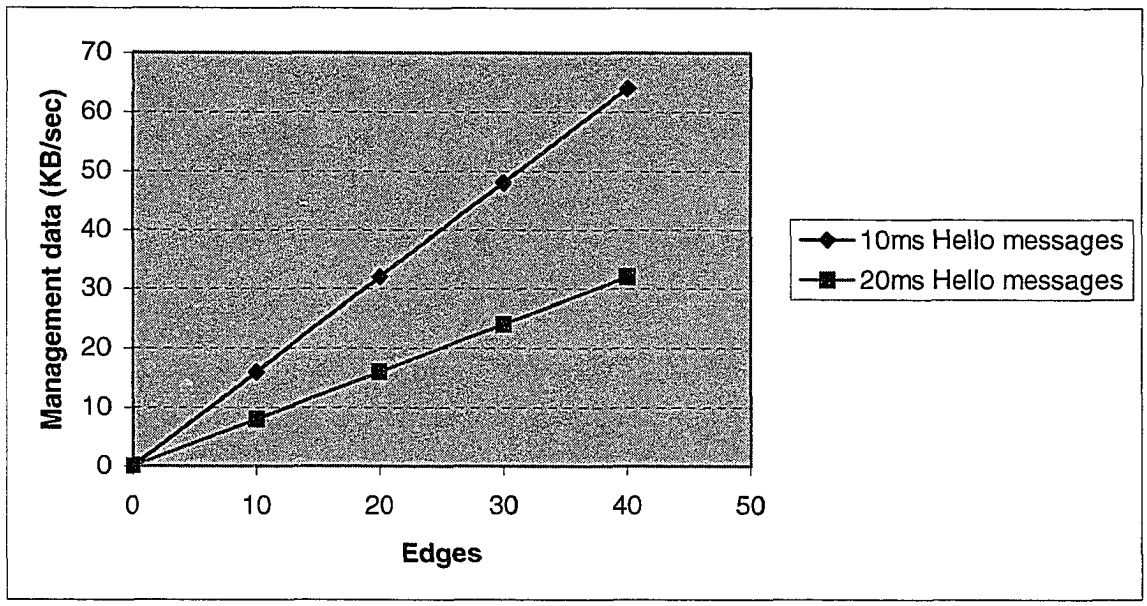


Figure 26: Management data transferred

Figure 26 shows the measured values. We note that significant amount of data is transferred between router and network manager when this option is chosen. Though in the base case, when there is a single router with no edges in the network, no management data need to be transmitted.

While the above graph is representative and used in choosing the design options, the following graphs are from the experiments conducted on the work done in this thesis.

6.2 End-to-End Performance

The experiment in computing the network topology refresh time at network manager is the sum of the following five delays.

- Time taken by network manager to create request for latest network topology from router.
- Time involved in passing the request from network manager to router.
- Time taken by the router to retrieve data from OSPF-TE database.
- Time involved in passing the response from router to network manager.
- Time taken by network manager to refresh the network topology after receiving latest topology from router.

Time involved in network manager creating the request for topology and sending the request to router are constant, given that only one network topology is requested at any time.

In the Figure 27, we provide the amount of time spent by the router in retrieving the network topology information from OSPF-TE database as a function of number of edges.

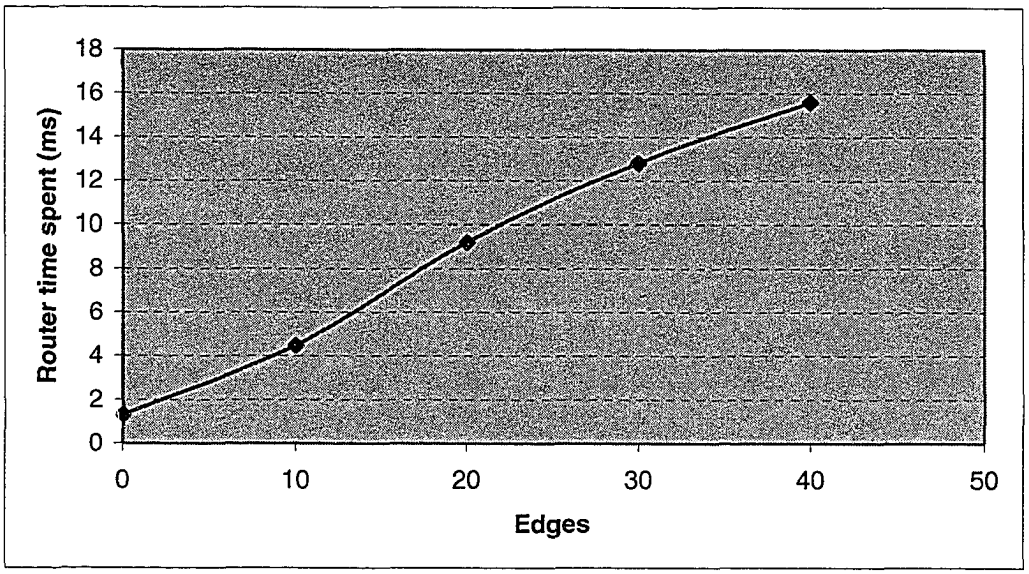


Figure 27: Router time spent in collecting topology information

In the base case with a single router with no edges in the network, the router still spends time in collecting the information in the empty LSDB. As seen in Figure 27, the router's work time to provide information to network manager is 1.3ms for base conditions and 15.6ms for 40 edges. However, this work to relay information can be done at the time router propagates information to other routers as part of the routing protocol update. Thus further minimizing the router's work time.

Next, communication time involved in sending management data from router to network manager as number of edges increase is given in Figure 28. It can be seen that as the edges increase, total elapsed time increases. Note that when the topology has no edges, still the base connection takes 18 ms and it takes 63.25ms for 40 edges.

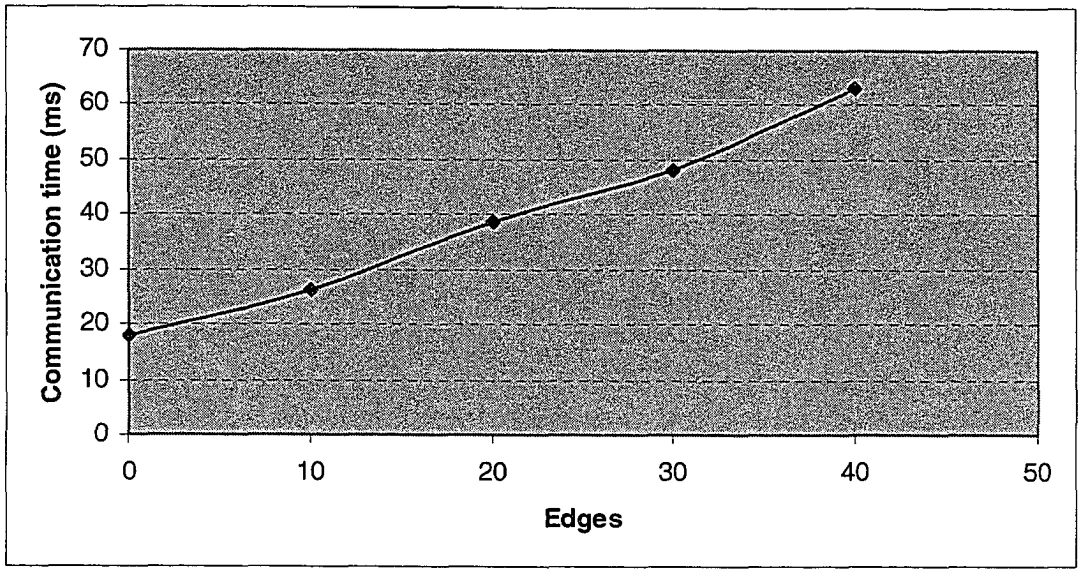


Figure 28: Communication time in sending management data

The measured amount of time taken by the network manager in refreshing the network topology after receiving the management data is measured and given in Figure 29. We can see that in base case, the network manager takes time to refresh a zero edge topology.

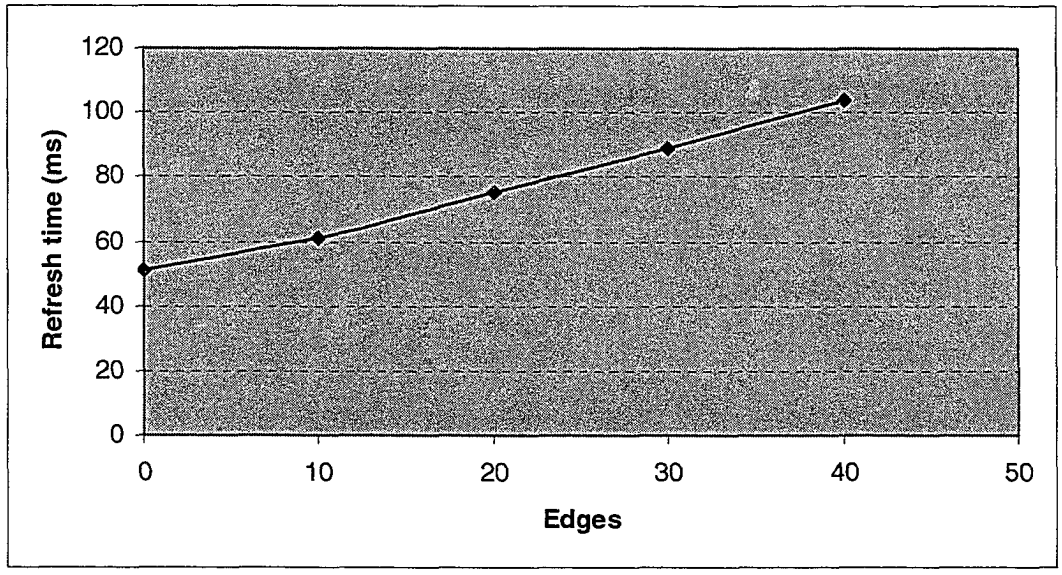


Figure 29: Network topology refresh time

In the Figure 30, the total amount of time taken by the network manager in refreshing network topology is given. The total time to refresh network topology is sum of the five delays described earlier in this section, namely:

- Time taken by network manager to create request for latest network topology from router.
- Time involved in passing the request from network manager to router.
- Time taken by the router to retrieve data from OSPF-TE database.
- Time involved in passing the response from router to network manager.
- Time taken by network manager to refresh the network topology after receiving latest topology from router.

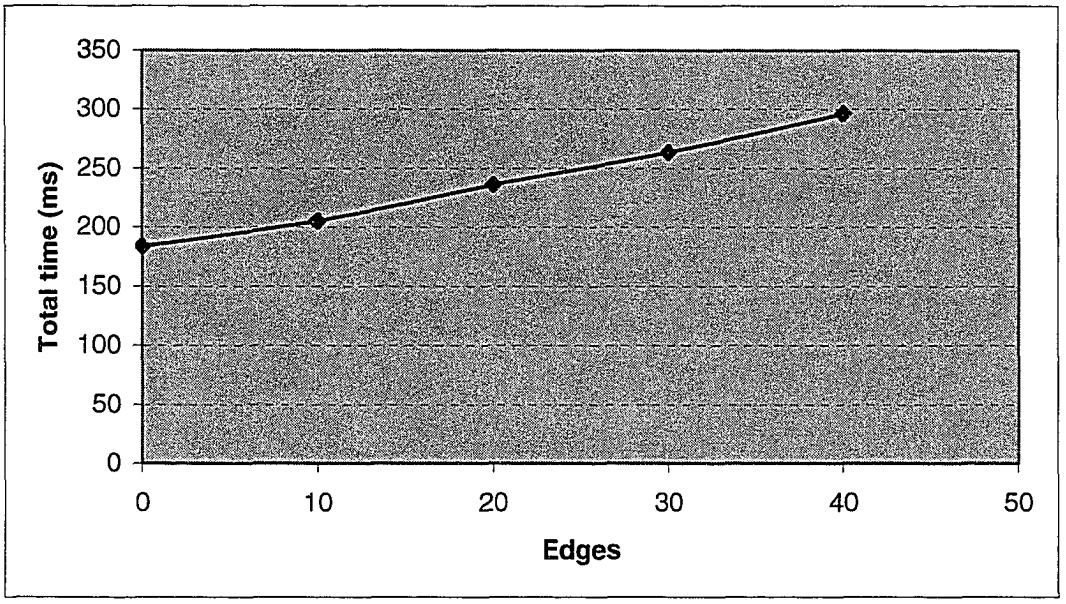


Figure 30: End-to-end total work time in the network manager

6.3 SPF/CSPF Real-time Calculations

The amount of time taken by CSPF versus the time taken by Dijkstra's shortest path first algorithm is studied next. This will give a clear idea on the extra time needed to handle constraints.

Amount of time required by CSPF = Time required to prune edges based on constraints + Dijkstra's SPF time.

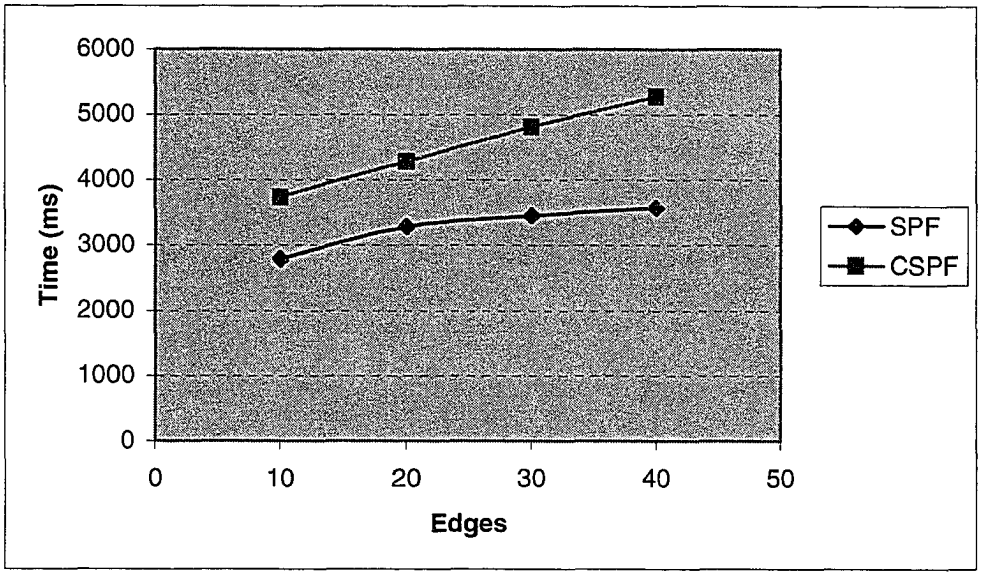


Figure 31: Path computation using SPF and CSPF

If the router provides information to network manager during regular updates to other routers then the additional router work time in Figure 27 specific to network manager is zero. However if network manager wants to get information for some reason, then maximum overhead on the router to provide this information is given in Figure 27 and it can be seen to be quite small (16ms for 40 edges).

If the network manager calculates LSP then Figure 31 shows the work time for CSPF to be in the range of 3736ms to 5280ms for 10 to 40 edges. Even if the router uses a 10 times faster embedded card, the time taken is not trivial. By moving the functionality to network manager, the network manager can calculate LSP offline. However, network manager needs the latest topology to calculate the LSPs. Figure 30 provides the time taken by network manager to get this latest topology from router. If we assume that router takes same time as network manager to calculate path, then the overhead of new architecture is 5% to 6% to achieve the separation (see Table 8). However the relief to router is quite large as seen in Figure 31.

#Edges	Extra time spent by network manager (ms)	Path calculation time (ms)	New architecture overhead (%)
10	204.98	3736	5.48
20	235.95	4276	5.51
30	263.25	4810	5.47
40	296.1	5280	5.6

Table 8: Overhead in using the new architecture

Therefore separating the path calculation functionality to a stand-alone system makes sense as:

- More intelligence could be added to calculate path
- Emulation could be done
- Overhead of accomplishing the separation is small
- Router work time is minimized

- Setting up of LSP need not be done in real-time, but does interfere with router execution affecting processing of other real-time application data.

6.4 Experimental Measurement Results

The experimental set up is shown in Figure 32. All the experimental values measured in milliseconds. We conducted the experiment over a router that has complete forwarding and routing components. The routing components consisted of OSPF-TE implementation. When the router was functional, we observed the background noise to be on an average 1.3% of CPU time. The background work items of the CPU are mostly maintenance tasks that were being run periodically. The worst-case background noise took 35.5% CPU utilization (in a 50 milliseconds time slot) and mostly remained at 0% in remaining time slots.

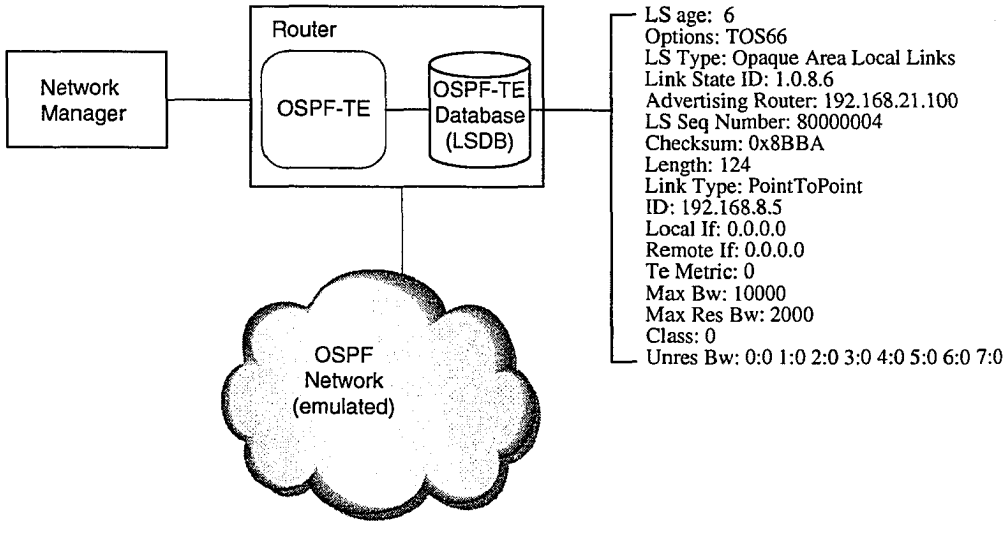


Figure 32: Experimental setup

For clarity purposes, we have given a sample LSA entry in Figure 33.

LSA age: 6
Options: TOS66
LS Type: Opaque Area Local Links
Link State ID: 1.0.8.6
Advertising Router: 192.168.21.100
LS Seq Number: 80000004
Checksum: 0x8BBA
Length: 124
Link Type: PointToPoint
ID: 192.168.8.5
Local If: 0.0.0.0
Remote If: 0.0.0.0
Te Metric: 0
Max Bw: 10000
Max Res Bw: 2000
Class: 0
Unres Bw: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0

Figure 33: LSA entry in LSDB

The experiments for analyzing end-to-end performance are divided into five parts. In the following, a detailed study of the measurement results is presented.

1. Time taken by network manager to create request to get latest network topology from router is constant (89.75ms).
2. Communication time involved in passing the request from network manager to router is also constant (23.5ms).
3. Time taken by the router to retrieve data from OSPF-TE database. The following router measurements are provided.

#Edges	Router time
0	1.3
10	4.48
20	9.2
30	12.8
40	15.6

Table 9: Router time to get network topology

4. Communication time involved in passing the response from router to network manager. For each case, five experiments are conducted.

#Edges	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average	Minimum	Maximum	Variance
0	16	18	19	18	19	18	16	19	1.2
10	26	26.25	24.8	26.5	27.7	26.25	24.8	27.7	0.866
20	38	37.5	39	40	38	38.5	37.5	40	0.8
30	47	46.5	49	52	46.5	48.2	46.5	52	4.46
40	65	66	63	61	61.25	63.25	61	66	3.95

Table 10: Communication time between router and network manager

The variance in experimental results is well within the bounds. The difference between maximum and minimum values is due to the background noise.

5. Time taken by network manager to refresh the network after receiving latest topology from router.

#Edges	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average	Minimum	Maximum	Variance
0	50	48.75	50.25	53	53	51	48.75	53	2.925
10	63	61.25	60	59	61.75	61	59	63	1.925
20	75.3	78.2	76	72	73.5	75	72	78.2	4.516
30	86.4	87.3	92	91.7	87.6	89	86.4	92	5.58
40	101.1	102	104.3	106.2	106.4	104	101.1	106.4	4.62

Table 11: Topology refresh time at network manager

From Table 11, it can be seen that the variance for topology refresh time at network manager is very small. The minimum and maximum values are very close.

The experimental measurements for analyzing SPF are shown in Table 12.

#Edges	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average	Minimum	Maximum	Variance
10	2786	2782	2782	2782	2788	2784	2782	2788	6.4
20	3273	3276	3271	3278	3277	3275	3271	3278	6.8
30	3455	3457	3449	3452	3457	3454	3449	3457	9.6
40	3573	3573	3568	3569	3567	3570	3567	3573	6.4

Table 12: SPF measurement results

CSPF experimental measurements are shown in Table 13.

#Edges	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5	Average	Minimum	Maximum	Variance
10	3732	3737	3733	3740	3738	3736	3732	3740	9.2
20	4274	4278	4280	4273	4275	4276	4273	4280	6.8
30	4807	4816	4811	4807	4809	4810	4807	4816	11.2
40	5282	5284	5279	5278	5277	5280	5277	5284	6.8

Table 13: CSPF measurement results

Table 12 and Table 13 show that the variance in computing the SPF and CSPF is small. The minimum and maximum values are tightly bound.

Chapter 7

Conclusions and Future Work

7.1 Summary

Network management and traffic engineering are related areas that deal with keeping good health of the networks and their efficient usage respectively. There are many proprietary solutions, which work closely with the network elements that are being managed. There are also a few generic solutions but they need significant amount of customization and are expensive for a small customer base with only a few network elements.

In this thesis, a viable network management and traffic engineering solution for a small network is provided. The major focus of the solution is displaying the current network configuration, managing the network elements using command line interface and web-based network management interface, and analyze what-if scenarios under different circumstances like node or link failure and with constraints like minimum bandwidth required along the path. In Chapter 1, routing in IP networks, its disadvantages and the importance of migrating to MPLS based networks are discussed. An introduction to network management and traffic engineering is also presented. Chapter 2 provided literature survey of network management and traffic engineering, different types of products available and their comparison. Chapter 3 laid the foundation for the problem addressed and why it is important to solve. Chapter 4 described the architectural options

explored and the preferred option for the problem. Chapter 5 explained the design and implementation of the solution. Chapter 6 provides performance analysis of the work.

7.2 Contributions of the Thesis

Traditional network management products offer some of the functionalities of FCAPS (fault, configuration, accounting, performance and security management). As the use of real-time applications is increasing, stringent QoS guarantees are required and mere FCAPS based management is insufficient. Network administrators have to be proactive in analyzing the available paths and suitably establish them. To address these concerns, in this thesis, an integrated network management system is designed and developed.

Network topology changes dynamically in terms of node or link availability and the level of congestion, and having the latest network topology is a constant concern for network administrators. It is essential to have a user interface, which displays the network topology with all relevant information about nodes and links. In this work, the network topology information is gathered in real-time from router and displayed at the network manager.

Another major thrust of this work is finding a suitable approach to analyze what-if scenarios for a network administrator. What-if scenarios are useful when a network manager wants to understand the network behavior even before some of the links get congested so that the traffic can be normalized to efficiently utilize the network. Also, this is a handy tool as it provides a way to establish optimal paths without violating a set of constraints like bandwidth. In this work, network administrator is provided with an

option to fetch the current topology and conduct what-if scenario analysis without utilizing network resources. Even some of the nodes and links can be taken down to emulate outage while analyzing the what-if scenarios.

To offer maximum flexibility in performing element management, an option is provided to invoke router CLI from router entity of the network topology. Web-based interface is built into the system to be able to access the network manager from any host in the network by using a web browser.

Third party routers running MPLS and OSPF-TE also can easily be managed by the integrated network management system with minimal interface on the router side.

The integrated network management system for MPLS networks is designed, developed and tested with real network devices.

7.3 Future Work

Although the solution architecture has sophisticated features, the options chosen are to make it inexpensive and efficient for a small customer network. As the situation demands, the other options can be explored to suit the needs. In the following, a few customization choices and good-to-add features are presented.

Customization choices to the solution presented in this thesis are:

1. In this thesis, network topology data is collected only from OSPF-TE database. However, when the network does not implement OSPF-TE or does not have access to

OSPF-TE database, network topology may be collected from other routing protocols like RIP, IS-IS or BGP as the network demands.

2. To offer a standards-savvy solution, standard network management protocols like SNMP may be used in place of the proprietary sockets approach used in this thesis.

Good-to-add feature additions to this solution are:

1. The solution uses what-if scenario analysis without reserving the path with Resource Manager. However, other options may be provided to reserve the paths with Connection Management so that there will not be any surprises when the path is set.
2. Currently, it is assumed that the network administrator provides the information of the nodes and links in the network manager database even before they were added to the network. This restriction may be removed and the node or link may be displayed with the available information.
3. On the network element side, as presented in Chapter 4, section titled Source of Network Topology Information, it is possible to allow the network administrator to reserve or take down some of the nodes or links so that they do not even appear in the network topology. Such a user interface component is very useful in a real network.
4. Additional traffic engineering constraints like delay and jitter can also be incorporated to get more appropriate LSPs.

References

[1] Bruce Davie, Yakov Rekhter, *MPLS Technology and Applications*, Morgan Kaufmann Publishers, Academic Press, 2000.

[2] Network Management System: Best Practices, *Cisco White Paper.* (http://www.cisco.com/warp/public/126/NMS_bestpractice.pdf, March 2005)

[3] Lakshmi Raman, OSI Systems and Network Management, *IEEE Communication Magazine*, March 1998, pp46-53.

[4] William Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, Third Edition, Addison Wesley, 1999.

[5] Stephen Morris, *Network Management, MIBs and MPLS: Principles, Design and Implementation*, Prentice Hall PTR, June 2003.

[6] Generalized Multi-Protocol Label Switching, *Alcatel Strategy White Paper.* (http://www.alcatel.com/doctypes/articlepaperlibrary/pdf/WP/S0312-GMPLS-EN.pdf, March 2005)

[7] A Comparison of Multiprotocol Label Switching (MPLS) Traffic Engineering Initiatives, Web ProForum Tutorials, *The International Engineering Consortium.* (http://www.iec.org, March 2005)

[8] Eric Osborne, and Ajay Simha, *Traffic Engineering with MPLS,* Cisco Press, July 2002.

[9] Overview of SNMP for the Cisco CMTS Router, *Cisco White Paper.* (http://www.cisco.com/univercd/cc/td/doc/product/cable/cmtsmib/cmtsmib1.pdf, March 2005)

[10] DR-Web Extensible Agent, *SNMP Research.* (http://www.snmp.com/products/drwebagt.html, March 2005)

[11] Hwa-Chun Lin and Chien-Hsing Wang, Distributed Network Management by HTTP-Based Remote Invocation, *IEEE Globecom*, 1999, pp1889-1893.

[12] HP OpenView Network Services Management Solution for MPLS Networks, *Hewlett Packard Technical Blueprint*, (http://www.openview.hp.com/solutions/nsm/index.html, March 2005)

[13] IBM Tivoli NetView, *IBM Product Information.* (http://www-306.ibm.com/software/tivoli/products/netview/, March 2005)

[14] VitalNet Network Performance Management Software, *Lucent Product Information.* (http://www.lucent.com/products/solution/0,,CTID+2020-STID+10439-SOID+1335-LOCL+1,00.html, March 2005)

[15] P. Flegkas, P. Trimintzios, G. Pavlou, I. Andrikopoulos, and C. F. Cavalcanti, On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks, *Proceedings of Workshop on Policies for Distributed Systems and Networks (Policy 2001)*, Spinger-Verlang LNCS-1995, Bristol, UK, January 2001. (http://www.ist-tequila.org/publications/policy_workshop-2001.pdf, March 2005)

[16] Joel Conover, Policy Based Network Management, *Network Computing*, November 29, 1999, pp1-14. (http://www.networkcomputing.com/1024/1024f1.html?ls=NCJS_1024bt, March 2005)

[17] What You Should Know Before Investing in Policy-Based Network Management, *White Paper prepared for Avaya*, October 2001. (http://www1.avaya.com/enterprise/whitepapers/AvayaWhitePaper.pdf, March 2005)

[18] Bruce Broadman, Orchestream Conducts PBNM with Precision, *Network Computing*, January 21, 2002, pp41-48. (http://img.cmpnet.com/nc/1302/graphics/1302f2_file.pdf?ls=NCJS_1302rt, March 2005)

[19] R-Series, *Intelliden - Network Management Case Study.* (http://www.intelliden.com/page.asp?id=Case_Studies&subID=Network_Management, March 2005)

[20] Redcell, *Dorado Software - Network Management Solutions.* (http://www.doradosoftware.com/html/solutions/solutions_net_mgmt.shtml, March 2005)

[21] Solstice Enterprise Manager, *Sun Microsystems - Product Information.* (http://wwws.sun.com/software/solstice/sem/, March 2005)

[22] Common Information Model (CIM) Standards, *Distributed Management Task Force.* (http://www.dmtf.org/standards/cim, March 2005)

[23] Microsoft Systems Management Server, *Microsoft - Product Information.* (http://www.microsoft.com/smserver/default.asp, March 2005)

[24] WMI – Windows Management Instrumentation, *Microsoft - Product Information.* (http://www.microsoft.com/whdc/system/pnppwr/wmi/default.mspx, March 2005)

[25] Phil Shafer, XML-based Network Management, *Juniper Networks White Paper.* (http://www.juniper.net/solutions/literature/white_papers/200017.pdf, March 2005)

[26] Special Issue on XML-Based Network Management, *IEEE Communications Magazine*, July 2004.

[27] Heather Kreger, Ward Harold, Leigh Williamson, *Java and JMX – Building Manageable Systems*, Addison Wesley, 2003.

[28] D. Awduche, A. Chiu, A. Elwalid, I. Widjala, and X. Xiao, Overview and Principles of Internet Traffic Engineering, *RFC 3272*, May 2002. (http://www.ietf.org/rfc/rfc3272.txt, March 2005)

[29] Manju Hegde, and Mort Naraghi-Pour, Engineering traffic in MPLS networks, *EE Times*, January, 2002. (http://www.eetimes.com/article/showArticle.jhtml?articleId=16503805, March 2005)

[30] Douglas Mauro, and Kevin Schmidt, *Essential SNMP,* O'Reilly, July 2001.

[31] Douglas E. Comer, *Internetworking with TCP/IP – Principles, Protocols and Architecture,* 3rd Edition, Prentice Hall, February 1998.

[32] J. Moy, OSPF Version 2, *RFC 2328*, April 1998. (http://www.ietf.org/rfc/rfc2328.txt, March 2005)

[33] D. Katz, K. Kompella, and D. Yeung, Traffic Engineering (TE) Extension to OSPF Version 2, *RFC 3630*, September 2003. (http://www.ietf.org/rfc/rfc3630.txt, March 2005)

[34] J. Boyle, V. Gill, A. Hannan, D. Cooper, D. Awduche, B. Christian, and W. S. Lai, Applicability Statement for Traffic Engineering with MPLS, *RFC 3346*, August 2002. (http://www.ietf.org/rfc/rfc3346.txt, March 2005)

[35] Graphical Layout Tool Kit, Tom Sawyer Software. (http://www.tomsawyer.com, March 2005)

[36] Cisco Any Transport over MPLS, *Cisco Overview.* (http://www.cisco.com/warp/public/cc/so/neso/vpn/unvpnst/atomf_ov.htm, March 2005)

[37] Virtual Leased Line Services Using Cisco MPLS DiffServ-Aware Traffic Engineering, *Cisco White Paper.* (http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/msdvl_wp.htm, March 2005)

[38] F. Le Faucheur, and W. Lai, Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering, *RFC 3564*, July 2003. (http://www.ietf.org/rfc/rfc3564.txt, March 2005)

[39] John Shapley Gray, *Interprocess Communications in UNIX – The Nooks & Crannies*, Second Edition, Prentice Hall, 1998.

[40] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, Second Edition, McGraw-Hill, 2001.

[41] Aika Pras, Bert-Jan van Beijnum, and Ron Sprenkels, Introduction to TMN, CTIT Technical Report 99-09, University of Twente, April 1999. (http://www.simpleweb.org/tutorials/tmn/tmn.pdf, March 2005)