DiffServ/MPLS Network Design and Management

A Thesis
Presented to
The Academic Faculty

by

Tricha Anjali

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
March 2004

DiffServ/MPLS Network Design and Management

Approved by:

Professor Ian F. Akyildiz, Advisor	Professor George Riley
Professor Mostafa H. Ammar (College of Computing)	Professor Raghupathy Sivakumar
Professor Chuanyi Ji	

Date Approved: March 30, 2004

To my parents,

Deepa Goel and Jia Lal Goel,

and to my husband,

Raghu Venkat

Acknowledgements

The completion of this dissertation would not have been possible without the encouragement, help and friendship of many individuals. It is my privilege to thank the people who have supported and guided me throughout my pursuit of higher education, and my sincere apologies to the people I may miss.

For lack of better words, I would have to use the clichéd term “friend, philosopher and guide” for describing my advisor, Prof. Ian F. Akyildiz. Under his guidance, I have learnt to identify and approach research problems, and to develop and present the solutions in a comprehensible manner. It was not all about research and publishing papers with him. He taught me the real meaning of “networking” and to work as a team. He is my role-model for an excellent researcher, mentor, advisor and teacher.

I would also like to acknowledge Dr. Mostafa H. Ammar, Dr. George Riley, Dr. Chuanyi Ji and Dr. Raghupathy Sivakumar for serving on my dissertation committee. Their insightful feedback and approval of my research goals and objectives have helped me orient my efforts towards a well-formulated thesis. I would like to thank Dr. Chuanyi Ji, in particular, for the innovative discussions during the preparation of the NSF proposals.

I would like to thank George Uhl from the NASA Goddard Space Flight Center for his financial support of the project and the constant feedback about the progress of the project. Also, I would like to thank Agatino Sciuto who joined the project at NASA for a brief interval. I also want to thank the National Science Foundation ITR program for the sponsorship of my research.

I would like to extend my sincere gratitude towards Dr. Caterina Scoglio in my research lab for her constant support, friendship and advice. Without her, we would not have been able to secure the financial support for funding this research. She kept the morale of the

IP QoS group high. Special thanks to Jaudelice C. de Oliveira (Jau) who, with her lively enthusiasm and thoughtful advice, helped me a lot with the research. I would also like to thank Benjamin King and Leonardo Chen who helped with the setup of the testbed and the implementation of the proposed policies on the testbed.

I would also like to thank the past and present members of the Broadband and Wireless Networking laboratory for their friendship and advice. Special thanks go to Janise Y. McNair, Wenye Wang, Eylem Ekici, and Weilian Su.

I would like to mention the systems support provided by Didier Contis. He helped with the procurement and setup of the testbed equipment. I thank Cordai Farrar, Kesha Jackson and Tina Clonts of the ECE department for their administrative assistance.

Finally, I would like to thank my parents for their love, support and encouragement. They provided me with an environment that supported my academic dreams. Lastly, I would like to thank my husband for his love, patience, support, counsel and sacrifices. He was my pillar of strength during the completion of this dissertation.

Table of Contents

List of Tables

List of Figures

Abbreviations

ABEst	Available Bandwidth Estimation
AF	Assured Forwarding
AS	Autonomous System
BB	Bandwidth Broker
BE	Best Effort
CMIP	Common Management Information Protocol
COPS	Common Open Policy Service
CR-LSP	Constraint-Based Routed LSP
CT	Class-Type
CTMDP	Continuous Time Markov Markov Decision Process
DiffServ	Differentiated Services
DS-TE	DiffServ Traffic Engineering
DSCP	Differentitated Services Codepoint
EF	Expedited Forwarding
EPAT	Estimation and Prediction Algorithm for TEAM
ER	Edge Router
FEC	Forwarding Equivalence Class
GMPLS	Generalized Multiprotocol Label Switching
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITE	Internet Traffic Engineering
LAN	Local Area Network

LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switching Router
MABE	Method for Available Bandwidth Estimation
MDP	Markov Decision Process
MIB	Management Information Base
MPET	Measurement/Performance Evaluation Tool
MPLS	MultiProtocol Label Switching
MRTG	Multi Router Traffic Grapher
NLANR	National Laboratory for Applied Network Research
OID	Object Identifier
PHB	Per-Hop Behavior
QoS	Quality of Service
RRD	Round Robin Database
RSVP	Resource Reservation Protocol
SLA	Service Level Agreement
SLS	Service Level Specification
SNMP	Simple Network Management Protocol
ST	Simulation Tool
TE	Traffic Engineering
TEAM	Traffic Engineering Automated Manager
TEMB	Tool for End-to-end Measurement of Available Bandwidth
TET	Traffic Engineering Tool
TOS	Type Of Service

Summary

The MultiProtocol Label Switching (MPLS) framework is used in many networks to provide efficient load balancing which distributes the traffic for efficient Quality of Service (QoS) provisioning in the network. If the MPLS framework is combined with Differentiated Services (DiffServ) architecture, together they can provide aggregate-based service differentiation and QoS. The combined use of DiffServ and MPLS in a network is called DiffServ-aware Traffic Engineering (DS-TE). Such DiffServ-based MPLS networks demand development of efficient methods for QoS provisioning. In this thesis, an automated manager for management of these DiffServ-based MPLS networks is proposed. This manager, called Traffic Engineering Automated Manager (TEAM), is a centralized authority for adaptively managing a DiffServ/MPLS domain and it is responsible for dynamic bandwidth and route management. TEAM is designed to provide a novel and unique architecture capable of managing large scale MPLS/DiffServ domains without any human interference. TEAM constantly monitors the network state and reconfigures the network for efficient handling of network events. Under the umbrella of TEAM, new schemes for Label Switched Path (LSP) setup/tear-down, traffic routing, and network measurement are proposed and evaluated through simulations. Also, extensions to include Generalized MPLS (GMPLS) networks and inter-domain management are proposed.

As a part of TEAM, an optimal threshold-based policy for LSP and setup/tear-down is proposed. It provides an on-line design for the MPLS network depending on the traffic load. The proposed policy is a traffic-driven approach and balances the bandwidth, switching and signaling costs for the network. Whenever a new connection request arrives, a decision is made whether to setup a new LSP, re-dimension a pre-existing LSP or route the traffic on a simple hop-by-hop IP route. A sub-optimal method is proposed that is easier to implement in actual networks. For the traffic request routing in DiffServ/MPLS networks, a new QoS routing algorithm is proposed. No stochastic model is assumed for the incoming traffic. Many QoS metrics such as distance, available bandwidth and delay constraints are considered before the path selection. Only partial knowledge about the network state is assumed at each network node. The algorithm finds a feasible path that minimizes the cost incurred. The cost is attributed to bandwidth carriage, switching and signaling efforts in the network. For the network measurement, various methods for measurement of available bandwidth in the network are proposed. Measurements are necessary to determine the network condition and performance of the various algorithms implemented by TEAM. Algorithms for measurement of the available bandwidth on a link and end-to-end path are presented. The link available bandwidth measurement algorithm predicts the duration for which the measure is valid with a high degree of confidence. The algorithm dynamically changes the number of past samples that are used for prediction and also the duration for which the prediction holds. Also, an algorithm for measurement of end-to-end available bandwidth is presented. The algorithm combines the advantages of both active and passive measurement methodologies to obtain accurate, reliable measurements of the available bandwidth along a path. It is based on probing the devices on the path to get information about the path statistics. Then, an algorithm is proposed for traffic estimation and resource allocation forecast on inter-domain links in an attempt to extend the manager for inter-domain operations. The algorithm is based on measurement of the current usage of the link. The algorithm allows efficient resource utilization while keeping the number of reservation modifications to low values. Finally, TEAM implementation details are provided with experimental results to demonstrate the performance of TEAM as an efficient network manager. As an attempt to extend TEAM functionality to the underlying optical network, an optimal policy has been proposed for the combined setup decision of LSPs and λSPs. This policy decides whether or not a direct λSP is beneficial to the network operational costs.

Chapter 1

Introduction

The Internet has evolved over time from a small set of interconnected computers in one room to a worldwide infrastructure. However, this growth has largely been uncoordinated. Networks and domains have been developed and deployed individually without end-to-end performance considerations. With this decentralized architecture, providing end-to-end service guarantees to applications is difficult. Nevertheless, the Internet users are developing new applications and expect better service from the Internet. Thus, the Internet architecture needs to be modified for delivering satisfactory service to the users.

1.1 Background

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Over the years, the IETF has proposed and developed several novel architectures for Quality of Service (QoS) provisioning in the Internet. Prominent among these are Integrated Services (IntServ), Differentiated Services (DiffServ) and MultiProtocol Label Switching (MPLS).

The IntServ architecture [1] has been designed to provide several classes of service in the network. The level of QoS provided by these enhanced QoS classes is programmable on a per-flow basis according to requests from the end applications. The requests dictate the level of resources (*e.g.* bandwidth, buffer space), called as “flowspec” [2], that must be reserved along with the transmission scheduling behavior that must be installed in the routers to provide the desired end-to-end QoS commitment for the data flow. These requests are passed to the routers using the Resource Reservation Protocol (RSVP) [3]. Once

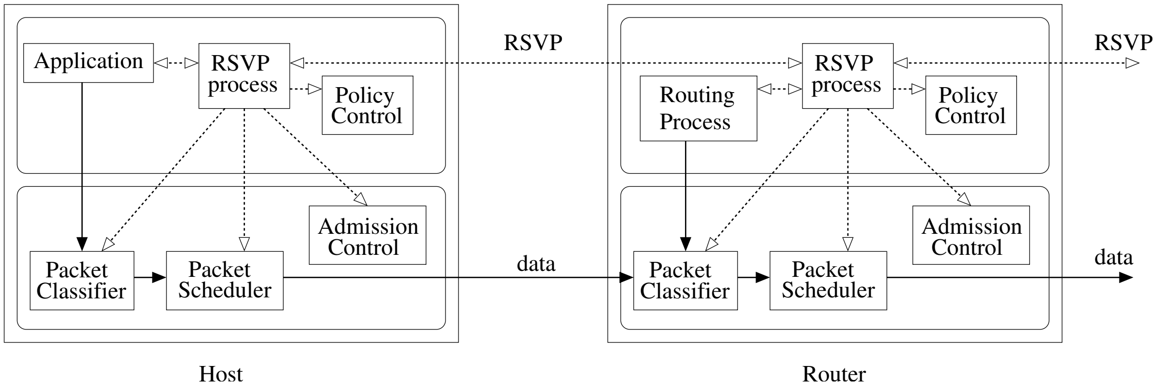


Figure 1: The IntServ Architecture.

an appropriate reservation has been installed in each router along the path, the data flow can expect to receive an end-to-end QoS commitment provided no path changes or router failures occur during the lifetime of the flow, and provided the data flow conforms to the traffic envelope supplied in the request. Service-specific policing and traffic reshaping actions are employed in the network to ensure that non-conforming data flows do not affect the QoS commitments for well-behaving data flows. The IETF has formally specified Guaranteed Service [4] and Controlled-Load Service [5] for use with RSVP [6]. The IntServ functionality in the host and network routers is shown in Figure 1. A survey of the IntServ architecture can be found in [7, 8].

However, the IntServ architecture is not appropriate for the Internet because of the inherent drawbacks in its design. The architecture is not scalable for large networks since it requires per-flow state to be maintained at each network node. Also every packet has to be classified into the different service classes. The on-demand reservations in the network for each flow introduce a high degree of complexity in the network nodes. For this reason, this architecture is hard to deploy in a real network.

In order to solve the scalability and flexibility related drawbacks of the IntServ architecture, the IETF proposed the DiffServ [9] model. The goal of the DiffServ framework is to provide a means of offering a spectrum of services in the Internet without the need for per-flow state and signaling in every router. By carefully aggregating a multitude of QoS-enabled flows into a small number of aggregates that are given a small number of differentiated treatments within the network, DiffServ eliminates the need to recognize and store information about each individual flow in core routers. Each DiffServ flow is policed and marked at the first trusted downstream router according to a contracted service profile, after which the flow is mingled with similar DiffServ traffic into an aggregate. All subsequent forwarding and policing is performed on the aggregates. The packets are marked by designating the ”Per-hop Behavior” (PHB) that packets are to receive by setting a few bits in the Internet Protocol (IP) v4 header Type Of Service (TOS) octet [10]. In this mapping, the first 6 bits, called the Differentiated Services Codepoint (DSCP), define the PHB. The PHBs are expected to be simple and they define forwarding behaviors that may suggest, but do not require, a particular implementation or queuing discipline. In addition to DiffServ-enabled packet forwarders, the network also requires classifiers, policers, markers and a new kind of network component known as a bandwidth broker [11]. Per-flow policing and marking is performed by the first trusted leaf router downstream from the sending host. When a local admissions control decision has been made by the sender’s cloud, the leaf router is configured with the contracted per-flow service profile. Downstream from the first leaf router, all traffic is handled as aggregates. Network domains may need to shape on egress to prevent otherwise conforming traffic from being unfairly policed at the next downstream domain. On domain ingress, incoming traffic is classified by the PHB bits into aggregates, which are policed according to the aggregate profiles in place. Depending on the particular DiffServ service model in question, out-of-profile packets are either dropped at the edge or are remarked with a different PHB. Finally, to make appropriate internal and external admissions control decisions and to configure leaf and edge device policers correctly, each domain is outfitted with a bandwidth broker (BB). Currently, two PHBs have been proposed by the IETF, namely Assured Forwarding (AF) [12], and Expedited Forwarding (EF) [13]. The DiffServ functionality in the network edge and core router is

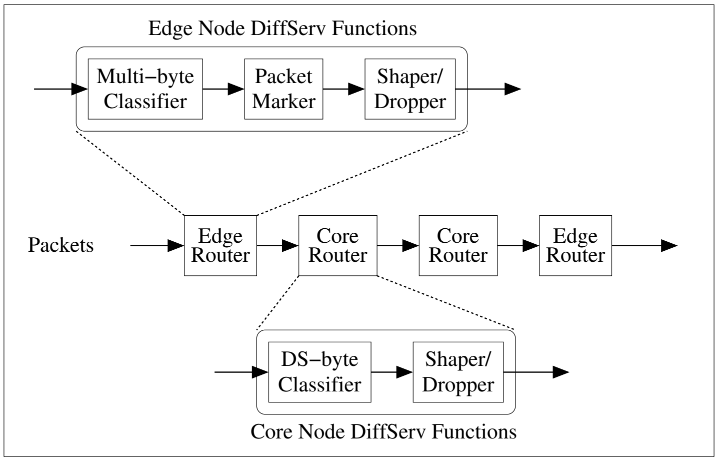


Figure 2: The DiffServ Architecture.

shown in Figure 2. A survey of the DiffServ architecture can be found in [14].

Although DiffServ solves the scalability problem of IntServ, it suffers from the so-called *resource stealing* drawback. Flows sharing a common class will compete inside the class for the resources available to all members and in some occasions might reduce the performance of their competitors in terms of QoS measures by stealing the resources that were initially used by their rivals. Unfortunately, the DiffServ standard does not propose a technique to alleviate the problem.

In an effort to combine the virtues of both IntServ and DiffServ, [15] proposes the operation of IntServ over DiffServ networks with the addition of flow admission control capabilities to the edge nodes of a statically provisioned DiffServ network region. This approach provides scalability to a large number of flows and strict service guarantees simultaneously. RSVP messages are exchanged end-to-end across the network. At the edge of the network, these messages are interpreted on a hop by hop basis as in standard IntServ. However, pure DiffServ regions can also be used within the path. Here, the RSVP messages are only interpreted at the edge of the DiffServ region. These nodes map RSVP reservation requests to DiffServ forwarding classes, keeping a record of the session identifier and the required DiffServ class. Once data begins to flow using the reservation, these edge nodes will map the packet headers to session identifiers in the packets and confirm that the correct DSCP is being used. Once within the network, all packets are forwarded according to standard DiffServ operation. This architecture still does not solve the DiffServ related issues of sub-optimal network resource usage and the cross interaction between traffic flows with different ingress/egress node pairs.

In an effort to alleviate the drawbacks of DiffServ, IETF proposed to use the MPLS technology in conjunction with DiffServ. The objective of MPLS is to increase the efficiency of data throughput by optimizing packet processing overhead in the IP networks. The MPLS architecture is detailed in [16]. The MPLS technology uses a short fixed-length label to route packets in the network. The edge routers in the network, called the Label Edge Routers (LERs), attach this label to the packet. The core routers in the network, called the Label Switching Routers (LSRs), then route the packet based on the assigned label rather than the original packet header. The label assignments are based on the Forwarding Equivalence Class (FEC) of the packet, where packets belonging to the same FEC are assigned the same label and generally traverse through the same path across the MPLS network. An FEC may consist of packets that have common ingress and egress nodes, or same service class and same ingress/egress nodes, etc. A path traversed by packets in the same FEC is called a Label Switched Path (LSP). The Label Distribution Protocol (LDP) and an extension to the Resource Reservation Protocol (RSVP) are used to establish, maintain (refresh), and tear-down LSPs [17]. MPLS performs a much faster forwarding than IP since the packet headers do not need to be analyzed at every hop in the path. MPLS also provides Traffic Engineering (TE) [18] by allowing traffic to be explicitly routed in the network to achieve efficient load balancing [19]. The requirements for Traffic Engineering over MPLS are given in [20]. The MPLS architecture in a network node is shown in Figure 3. Details about the MPLS architecture can be found in [21].

Generalized MPLS (GMPLS) is an extension to MPLS as a control plane solution for next generation optical networking. It enables Generalized Label Switched Paths (G-LSPs)

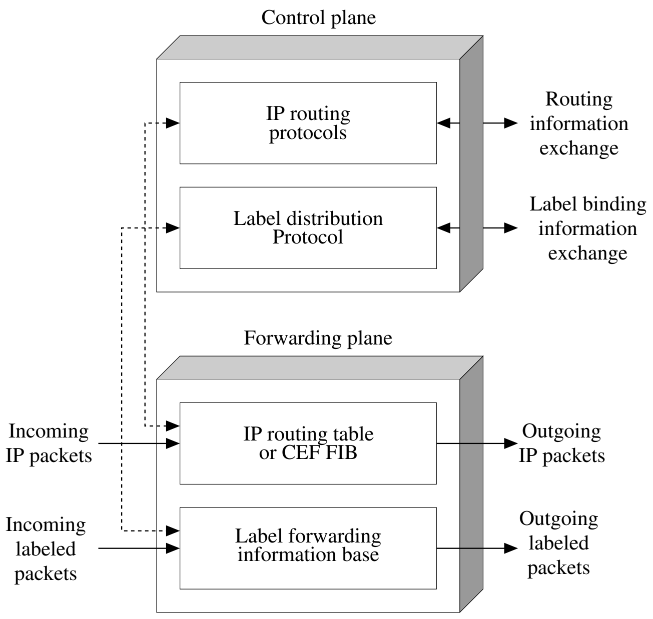

Control plane
IP routing protocols
Label distribution Protocol
Routing information exchange
Label binding information exchange
Forwarding plane
Incoming IP packets
IP routing table or CEF FIB
Outgoing IP packets
Incoming labeled packets
Label forwarding information base
Outgoing labeled packets

Figure 3: The MPLS Architecture.

such as lightpaths [22], to be automatically setup and torn down by means of a signaling protocol [23]. GMPLS differs from traditional MPLS because of its added switching capabilities for lambda, fiber etc. It is the first step towards the integration of data and optical network architectures. It reduces network operational costs with easier network management and operation. The traditional MPLS is defined for packet switching networks only. MPLS mainly focuses on the data plane as opposed to GMPLS' focus on control plane. GMPLS extends the concept of LSP setup beyond the Label Switched Routers (LSRs) to wavelength/fiber switching capable systems. Thus, GMPLS allows LSP hierarchy (one LSP inside another) at different layers in the network architecture.

To achieve fine-grained optimization of transmission resources and further enhanced network performance and efficiency, traffic engineering must be performed at a per-class level [24] in the MPLS network. Thus, DiffServ mechanisms [25] may be used to complement the MPLS TE mechanisms [20, 26, 27] because they operate on an aggregate basis across all DiffServ classes of service. In this case, DiffServ and MPLS TE both provide

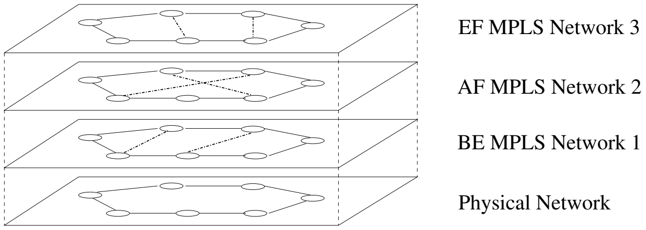


Figure 4: Virtual MPLS Networks.

their respective benefits. By mapping the traffic from different DiffServ classes of service onto separate MPLS LSPs, DiffServ-aware MPLS networks can meet engineering constraints which are specific to the given class on both shortest and non-shortest path. This TE strategy is called DiffServ-aware Traffic Engineering (DS-TE) [24] and currently, three class types are defined for different DiffServ PHBs. Traffic belonging to each class type is carried on a virtual MPLS network by itself. These MPLS networks are layered on top of the physical network as shown in Figure 4. Each physical link capacity is partitioned among different MPLS networks and a maximum capacity is assigned to each partition according to a bandwidth constraint model. Many constraint models are under development at the IETF at the time of writing of this dissertation, such as russian doll, maximum allocation, maximum allocation with reservation etc. The unused reserved bandwidth is then used for Best Effort (BE) traffic. The design and management of these over-layered MPLS networks is a fundamental key to the success of the DiffServ-MPLS mapping. These networks should be managed independently to consider the QoS requirements of each traffic class. Many open research issues need to be solved for efficient management of these networks, such as LSP dimensioning, setup/tear-down, routing, preemption, initial definition of the network topology, etc.

1.2 Research Objectives and Related Work

In this thesis, new techniques for the design and management of these virtual MPLS networks are proposed and developed. In particular, the following areas are investigated:

1. Automated network manager

2. LSP and setup and tear-down

3. Traffic routing

4. Link/LSP available bandwidth estimation

5. End-to-end available bandwidth measurement

6. Inter-domain management

7. Optical network topology design

Automated network manager

Few comprehensive traffic engineering managers have been proposed in literature. The Routing and Traffic Engineering Server (RATES) [28] is a software system developed at Bell Laboratories for MPLS traffic engineering, but TE is only performed for the routing of bandwidth guaranteed LSPs. MPLS Adaptive Traffic Engineering (MATE) [29] is another state dependent traffic engineering mechanism for distributing network load adaptively. MATE assumes that several explicit LSPs have been established between ingress and egress nodes in an MPLS domain using a standard protocol like RSVP-TE. MATE is suitable when only a few ingress-egress pairs are considered and it is not designed for bandwidth guaranteed services. Traffic Engineering for QUality of service in the Internet, at LArge scale (TEQUILA) [30, 31] is a European collaborative research project looking at an integrated architecture and associated techniques for providing end-to-end QoS in a DiffServ-based Internet. Although the TEQUILA architecture is very interesting, the algorithms and techniques to be implemented in TEQUILA are not defined in detail at the moment, and their quantitative evaluation has not been carried out. GlobalCrossing decided to use MPLS for TE, QoS and Virtual Private Network (VPN) provisioning [32]. In

[33], the authors have demonstrated that the MPLS traffic engineering has been effective in meeting the delay and jitter bounds required by applications.

In this thesis, an automated manager for DiffServ/MPLS networks is introduced. The Traffic Engineering Automated Manager (TEAM) is comprised of a central server, the Traffic Engineering Tool (TET), that is supported by two additional tools: the Measurement/Performance Evaluation Tool (MPET) and the Simulation Tool (ST). The TET and the MPET interact with the routers and switches in the domain. The MPET provides a measure of the various parameters of the network and routers. This information is input to the TET. Based on this measured state, the TET performs the resource and route management in the network. The TET also automatically implements the action, configuring accordingly the routers and switches in the domain. Whenever required, the TET can consolidate the decision using the ST. The ST simulates a network with the current state of the managed network and applies the decision of the TET to verify the achieved performance. The TET management tasks include Bandwidth Management (LSP setup/dimensioning, LSP preemption, LSP capacity allocation) and Route Management (LSP routing, traffic routing). Details of the architecture and implementation are described.

LSP setup and tear-down

Much of the research effort in the MPLS networks is based on the assumption that the request for the setup of an LSP is received by the ingress LSR from a Service Level Agreement (SLA) for the traffic, the LSP is manually setup and the traffic then utilizes the LSP. Very few research proposals have addressed the traffic-driven LSP setup and tear-down decision problem. One such traffic-driven LSP setup policy has been proposed in [34], in which an LSP is established whenever the number of bytes forwarded within one minute exceeds a certain amount. The proposed policy reduces the number of LSPs in the network; however, it has very high signaling costs and needs high control efforts for variable and bursty traffic in the case of a fully connected network. Another traffic driven LSP setup method for multicast in an MPLS network is described in [35].

In this thesis, a new optimal traffic-driven decision policy is introduced to determine and adapt the MPLS network topology based on the current traffic load. The objective of the proposed policy is to minimize the costs involving bandwidth, switching and signaling. The policy is derived by utilizing the Markov Decision Process theory. The policy decides when to setup a direct LSP, when to re-dimension an existing one, how to route traffic and also when to tear-down the LSP. In addition to the optimal policy, a sub-optimal policy is also proposed which is less computationally intensive but has comparable performance to the optimal policy. Since the traffic load may change depending on time, the new policy performs filtering in order to avoid oscillations which may occur in case of variable traffic.

Traffic routing

Routing is an extensively studied subject [36]. It has come a long way from the simple Dijkstra routing [37]. Much of the work in the field of QoS routing has concentrated on the delay constrained least cost problem [38, 39, 40]. Since the problem is NP-complete, the proposed solutions are heuristic in nature [41]. Some effort has also concentrated towards heuristic algorithms based on Lagrangian relaxation [42, 43]. This approach does not have the capability to consider non-additive metrics for the route computation. In MPLS networks, the routing research has concentrated on LSP routing *i*.e. how to route the LSPs in the network. Many schemes such as Minimum Interference Routing Algorithm (MIRA) [44], Profile Based Routing (PBR) [45] have been proposed for LSP routing. However, a scheme for routing of traffic flows in an MPLS network is not considered.

In this thesis, a QoS traffic routing algorithm that considers multiple metrics, is scalable and operates in the presence of inaccurate information, is presented. This routing algorithm is unique because of the dynamic nature of the MPLS network topology. Numerous path choices are compared in terms of their operational costs. The cost considers all the metrics important for the path selection. The factors pertaining to the different metrics are weighed by their corresponding importance factor which can be varied from network to network. In essence, the novelty of the proposed algorithm lies in the cost structure for the LSPs and the ability to deal with the partial network state information. The proposed routing algorithm also uses a prediction procedure at the network nodes to deal with the partial information available at nodes for scalability concerns.

Available bandwidth estimation

The available bandwidth on a link is indicative of the amount of load that can be routed on the link. Obtaining an accurate measurement of the available bandwidth is crucial to effective deployment of QoS services in a network. In [46], the authors have described a few bottleneck bandwidth algorithms. They can be split into two families: those based on pathchar [47] algorithm and those based on Packet Pair [48] algorithm. In [49], the authors have proposed another tool to measure bottleneck link bandwidth based on packet pair technique. Some other tools based on the same technique for measuring bottleneck bandwidth of a route have been proposed in [50, 51]. None of them measures the available bandwidth or utilization of a desired link of a network. In [52], the authors have proposed a tool to measure the available bandwidth of a route which is the minimum available bandwidth along all links of the path. It is an active approach based on transmission of self-loading periodic measurement streams. Another active approach to measure a path's available capacity is given in [53]. Iperf [54] from National Laboratory for Applied Network Research (NLANR) is another active approach that sends streams of TCP/UDP flows. Cisco has introduced the NetFlow [55] technology that provides IP flow information for a network. However, a network manager such as TEAM requires a tool for measuring the available bandwidth on a certain link of the network in a passive manner whenever desired.

In this thesis, an algorithm is presented to estimate the available bandwidth of a network link. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. The algorithm dynamically changes the number of past samples that are used for prediction and also the duration for which the prediction holds. The approach is based on the use of Multi Router Traffic Grapher (MRTG) where TEAM enquires each router in the domain through SNMP and obtains the information about the available bandwidth on each of its interfaces.

End-to-end available bandwidth measurement

The first tool that attempted to measure available bandwidth was cprobe [56]. This tool estimated the available bandwidth based on the dispersion of long packet trains at the receiver. A similar approach was given in pipechar [57]. The underlying assumption for these tools is that the dispersion of long packet trains is inversely proportional to the available bandwidth. However, this is not true [58]. Another measurement technique, Delphi [59], assumes that the path can be well modeled by a single queue and so it is not applicable when there are significant queuing delays in several links of the path. In [60], a tool to measure the available bandwidth of a path is presented. It is an active approach based on transmission of self-loading periodic measurement streams. This scheme sends traffic at increasing rates from the source to the destination until the rate finally reaches the available bandwidth of the tight link after which the packets start experiencing increasing delay.

In this thesis, a tool is proposed for measuring end-to-end available bandwidth over a path that can possibly span across multiple domains. The tool is efficient, easy to implement, and a combination of active and passive approaches. This way, it derives the benefits of both the measurement approaches. The tool is designed such that the measurement packets are processed with about the same computation level as IP forwarding. It utilizes the interface information from the Management Information Base (MIB) in the routers along the path. The functionality of the tool is distributed between both the source and destination of the path whose measurement is desired. The source sends measurement packets that collect information along the path and are returned back by the destination to the source for analysis.

Inter-domain management

Neighboring TEAMs communicate with each other to establish resource reservation agreements. Conventional approaches for resource allocation on links rely on pre-determined traffic characteristics. Current resource allocation methods can be either *off-line* or *on-line*. Off-line, or static, methods determine the allocation amount before the transmission begins. These approaches (*e.g.* [61]) are simple and predictable but lead to resource wastage. On-line, or dynamic, methods (*e.g.* [62, 63, 64]) periodically renegotiate resource allocation based on predicted traffic behavior. These methods undergo a large number of renegotiations to satisfy the QoS. An on-line scheme for resource provisioning is to have a bandwidth "cushion", wherein extra bandwidth is reserved over the current usage. As proposed in [65], if the traffic volume on a link exceeds a certain percentage of the agreement level, it leads to a multiplicative increase in the agreement. A similar strategy is proposed in case the traffic load falls below a considerable fraction of the reservation. This scheme satisfies the scalability requirement but leads to inefficient resource usage which becomes increasingly significant once the bandwidth requirements of the users are considerable.

In this thesis, an on-line scheme to forecast the bandwidth utilization of inter-domain links is presented. The scheme is designed to be simple, yet effective, when compared to more advanced prediction algorithms. The first step of the scheme is to perform an optimal estimate of the amount of traffic, belonging to a given traffic class, utilizing an inter-domain link based on a periodic measurement of the instantaneous traffic load. This estimate is then used to forecast the traffic bandwidth requests so that resources can be provisioned between the two domains to satisfy the QoS of the requests. The estimation is performed by the use of Kalman Filter [66] theory while the forecast procedure is based on deriving the transient probabilities of the possible system states. This scheme outperforms the current resource reservation mechanism ("cushion-based" allocation [65, 67]) employed by domain managers and also some other prediction schemes based on Gaussian [68, 69] as well as local maximum [70] predictor.

Optical network topology design

For the optical network underlying the MPLS network, most of the research efforts have concentrated on off-line topology design [71, 72, 73, 74], based on traffic matrix assumptions. The few on-line approaches to optical network topology design [75] do not consider the simultaneous design of the optical and MPLS networks.

In this thesis, an optimal LSP and λSP setup policy is provided which is obtained by extending the prior LSP setup policy for the underlying optical network. The Integrated Traffic Engineering (ITE) paradigm provides mechanisms for dynamic addition of physical capacity to optical networks. In the absence of such mechanisms, the rejection of incoming requests will be higher. The objective of the proposed policy is to minimize the costs involving bandwidth, switching and signaling. The policy decides when to setup a direct λSP, re-dimension an existing one or to route a LSP over multi-λSP path. In addition to the optimal policy, a sub-optimal policy and a threshold policy are also proposed which are less computationally intensive but have comparable performance to the optimal policy.

1.3 Thesis Outline

Chapter 2 presents the framework of the Traffic Engineering Automated Manager for the MPLS/DiffServ network management. The architecture of TEAM is described. The individual policies and algorithms for the network management decisions are described in the following chapters. Chapter 3 presents an optimal policy for LSP setup and tear-down in MPLS networks. The policy takes into account the bandwidth, switching and signaling costs at the MPLS network level. Whenever a new connection request arrives, a decision is made whether to setup a new LSP, to re-dimension the pre-existing LSP or to route the traffic request on a simple hop-by-hop IP route. Chapter 4 introduces a QoS traffic routing algorithm that considers multiple metrics, is scalable and operates in the presence of inaccurate information. Three algorithms are described in increasing order of complexity, in their centralized and distributed versions. The paths are chosen based on their cost which considers various metrics important for the path selection such as link available bandwidth, delay etc. Chapter 5 presents an algorithm to estimate the available bandwidth on a network link. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. Chapter 6 proposes a tool for measuring end-to-end available bandwidth over a path that can possibly span across multiple domains. The tool is efficient, easy to implement, and a combination of active and passive approaches. The tool utilizes the interface information from the Management Information Base (MIB) in the routers along the path. Chapter 7 presents an on-line scheme to forecast the bandwidth utilization of inter-domain links, in an effort to extend the operation of TEAM for inter-domain management. The scheme is split into two steps. The first step performs an optimal estimate of the amount of traffic, belonging to a given traffic class, utilizing an inter-domain link based on a periodic measurement of the instantaneous traffic load. This estimate is then used to forecast the traffic bandwidth requests so that resources can be provisioned between the two domains to satisfy the QoS of the requests. After introducing these individual components of TEAM, Chapter 8 presents the implementation details of TEAM. The software code is described and experimental results are presented for the synergistic operation of the TEAM components. Finally, Chapter 9 completes the thesis with concluding remarks and a discussion of future work. As an extension to the LSP setup policy in Chapter 3, Appendix A presents the optimal decision policy for λSP setup in the optical networks. This policy decides how to route the LSP and whether a direct λSP is needed at the optical network level.

Chapter 2

Traffic Engineering Automated Manager (TEAM)

DiffServ/MPLS networks need to be managed efficiently for QoS provisioning. In this chapter, the *Traffic Engineering Automated Manager* (TEAM) for such DiffServ/MPLS networks is introduced. TEAM is composed of a Traffic Engineering Tool (TET), which deals with resource and route management issues, a Measurement and Performance Evaluation Tool (MPET), which measures important parameters in the network and inputs them to TET, and a Simulation Tool (ST), which may be used by TET to consolidate its decisions. The TEAM architecture was first introduced in [76], and was later revised in [77].

This chapter is organized as follows: An introduction to network management is provided in Section 2.1. The motivation for the development of the network manager is given in Section 2.2. In Section 2.3, other attempts at development of a comprehensive network manager are described. Then, in Section 2.4, the architecture of the TEAM framework is presented with descriptions of the Traffic Engineering Tool in Section 2.4.1, the Measurement/Performance Evaluation Tool in Section 2.4.2, and the Simulation Tool in Section 2.4.3.

2.1 Network Management

Efficient network management involves a distributed database, auto-polling of network devices, high-end workstations generating real-time graphical views of network topology and traffic. In general, network management is a service that employs a variety of tools, applications, and devices to assist human network managers in monitoring and maintaining networks. The International Organization for Standardization (ISO) network management model consists of five conceptual areas: performance, configuration, accounting, fault, and security management. The goal of performance management is to measure and make available various aspects of network performance so that inter-network performance can be maintained at an acceptable level. Examples of performance variables that might be provided include network throughput, user response times, and line utilization. The goal of configuration management is to monitor network and system configuration information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed. The goal of accounting management is to measure network utilization parameters so that individual or group uses on the network can be regulated appropriately. The goal of fault management is to detect, log, notify users of, and (to the extent possible) automatically fix network problems to keep the network running effectively. Finally, the goal of security management is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be accessed without authorization.

An essential component of network management is the Network Management Protocol which is used by the management agent to exchange management information. The two most common network management protocols are Simple Network Management Protocol (SNMP) [78] and Common Management Information Protocol (CMIP) [79]. SNMP is by far the most widely used network management protocol and its use is widespread in Local Area Network (LAN) environments. CMIP is used extensively in telecommunication environments, where networks tend to be large and complex. It uses an ISO reliable connection-oriented transport mechanism and has built in security that supports access control, authorization and security logs. CMIP's significant disadvantage is that the protocol takes more system resources than SNMP by a factor of ten. The largest advantage to using SNMP is that its design is simple, hence it is easy to implement on a large network, for it neither takes a long time to set up nor poses a lot of stress on the network. SNMP accesses the Management Information Base (MIB) [80] of the managed devices. A MIB is a collection of hierarchically organized information which is identified by object identifiers.

2.2 Motivation

The combined use of the Differentiated Services (DiffServ) and the MultiProtocol Label Switching (MPLS) technologies is envisioned to provide guaranteed Quality of Service (QoS) for multimedia traffic in IP networks, while effectively using network resources [18]. By mapping the traffic from different DiffServ classes of service on separate LSPs, DiffServ-aware MPLS networks can meet engineering constraints specific to the given class on both shortest and non-shortest path. This TE strategy is called DiffServ-aware Traffic Engineering (DS-TE). In [25], the authors suggest how DiffServ behavior aggregates can be mapped onto LSPs. Such DiffServ-based MPLS networks should not be managed manually, since the network needs to respond promptly to changing traffic conditions. Therefore, automated managers are needed to simplify network management and to engineer traffic efficiently [81].

With the objective to study and research the issues mentioned above, an IP QoS testbed composed of Cisco routers was assembled in the Broadband and Wireless Networking Laboratory (BWN-Lab). This testbed is a high-speed top-of-the-line mix of highly-capable routers and switches for testing DiffServ and MPLS functionalities. During experiments with the testbed (results of the experiments were presented in [82]), the need for an improved set of algorithms for network management and also an integrated architecture for an automated network manager was clear. This led to the design and implementation of TEAM, a Traffic Engineering Automated Manager. Individual problems addressed by TEAM have already been considered, but an integrated solution does not exist in the research field. TEAM is developed as a centralized authority for managing a DiffServ/MPLS domain and is responsible for dynamic bandwidth and route management. Based on the network states, TEAM takes the appropriate decisions and reconfigures the network accordingly. TEAM is designed to provide a novel and unique architecture capable of managing large scale MPLS/DiffServ domains.

TEAM addresses the following network management issues:

- *Resource Management*: new schemes were developed to dynamically setup and dimension LSPs, allocate their capacity based on traffic estimation, and to preempt low priority LSPs to accommodate new high priority LSPs depending on the actual load on the network.

- *Route Management*: new schemes were developed to route LSPs and forward packets on a state-dependent basis to meet the QoS requirements.

The integration of the above mentioned tools results in a valuable resource for a network manager, in order to provide QoS and better network resource utilization.

2.3 Related Work

Much effort has been concentrated in the literature on individual research topics which are parts of TEAM. For example, an approach for continuous tuning of the network based upon on-line modeling, parameter search, and simulation capabilities of a simulation system is given in [83]. Another approach for automated and software-intensive configuration management of network inventory is given in [84]. Architecture for the design and implementation of active nodes to support different types of execution environment, policy-based driven network management, and a platform-independent approach to service specification and deployment has been proposed in [85].

Few comprehensive traffic engineering managers have been proposed in literature, and furthermore they address only a subset of the issues covered by TEAM. The Routing and

Traffic Engineering Server (RATES) [28] is a software system developed at Bell Laboratories for MPLS traffic engineering and is built using centralized paradigm. RATES communicates only with the source of the route and spawns off signaling from the source to the destination for route setup. RATES views this communication as a policy decision and therefore uses Common Open Policy Service (COPS) protocol. RATES uses a relational database as its information store. RATES implements Minimum Interference Routing Algorithm (MIRA) [44] to route LSPs. It consists of the following major modules: explicit route computation, COPS server, network topology and state discovery, dispatcher, Graphical User Interface, an open Application Programming Interface, data repository, and a message bus connecting these modules. Summarizing, RATES is a well designed TE tool, but TE is only performed for the routing of bandwidth guaranteed LSPs.

Another state dependent traffic engineering mechanism for distribute network load adaptively is suggested in [29]. MPLS Adaptive Traffic Engineering (MATE) assumes that several explicit LSPs have been established between an ingress and egress node in an MPLS domain using a standard protocol like RSVP-TE. The goal of the ingress node is to distribute the traffic across the LSPs. It is important to note that MATE is intended for traffic that does not require bandwidth reservation with best-effort traffic being the most dominant type. Since the efficacy of any state-dependent traffic engineering scheme depends crucially on the traffic measurement process, MATE requires only the ingress and the egress nodes to participate in the measurement process. Based on the authors' experience, available bandwidth was considered difficult to be measured, so packet delay and loss have been selected for measurement purposes. The network scenario for which MATE is suitable is when only a few ingress-egress pairs are considered. In fact, for a network with N nodes and x LSPs between each pair of nodes, the total number of LSP is in the order of xN^2 which can be a large number. Furthermore, it is not designed for bandwidth guaranteed services.

Traffic Engineering for QUality of service in the Internet, at LArge scale (TEQUILA)

[31] is a European collaborative research project looking at an integrated architecture and associated techniques for providing end-to-end QoS in a DiffServ-based Internet. In TEQUILA, a framework for Service Level Specification has been produced, an integrated management and control architecture has been designed and currently MPLS and IP-based techniques are under investigation for TE. The TEQUILA architecture includes control, data and management planes. The management plane aspects are related to the concept of Bandwidth Broker (BB) and each Autonomous System should deploy its own BB. The BB includes components for monitoring, traffic engineering, SLS management and policy management. The TE subsystem is further decomposed into modules of traffic forecast, network dimensioning, dynamic route management, and dynamic resource management. The MPLS network dimensioning is based on the hose model which is associated with one ingress and more than one egress node. The dynamic route management module considers: a) setting up the forwarding parameters at the ingress node so that the incoming traffic is routed to LSPs according to the bandwidth determined by network dimensioning, b) modifying the routing according to feedback received from network monitoring and c) issuing alarm to network dimensioning in case available capacity can not be found to accommodate new connection requests. The dynamic resource module aims at ensuring that link capacity is appropriately distributed among the PHBs sharing a link, by appropriately setting buffer and scheduling parameters. TEQUILA architecture is very interesting and shows a similar approach for MPLS networks design and management compared to TEAM. However, the algorithms and techniques to be implemented in TEQUILA are not defined in detail at the moment, and their quantitative evaluation has not been carried out.

The use of MPLS for traffic engineering, quality of service provisioning and virtual private networks has been decided at GlobalCrossing [32]. Approximately 200 routers participate in the MPLS system. Since a full meshed network would result in an MPLS system of about 40,000 LSPs, it is decided to deploy a hierarchical MPLS system of two layers of

LSPs. To deploy an MPLS system for traffic engineering, the following procedure is proposed based on the network operator experience: a) Statistics collected for traffic utilizing LSPs, b) Deploy LSPs with bandwidth constraints, c) Periodic update of LSP bandwidth d) Off-line Constraint based routing. To provide Quality of Service, MPLS is used in combination with the DiffServ architecture. It is desirable to use different LSPs for different classes. The effect is that the physical network is divided into multiple virtual networks, one per class. These networks can have different topology and resources. The end effect is that premium traffic can use more resources. Many tools are needed for designing and managing these virtual networks. The use of MPLS for TE and QoS decided by an important Internet Service Provider (ISP) is the confirmation that MPLS is a very promising technique even from a business point of view. The solution provided by TEAM is in line with the QoS architecture defined by GlobalCrossing. In [33], the authors have demonstrated that the MPLS traffic engineering has been effective in meeting the delay and jitter bounds required by applications.

2.4 TEAM Architecture Description

The proposed architecture of the TEAM is shown in Figure 5. TEAM has a central server, the Traffic Engineering Tool (TET), which is supported by two additional tools: Simulation Tool (ST) and Measurement/Performance Evaluation Tool (MPET). The TET and the MPET interact with the routers and switches in the domain. The MPET provides a measure of the various parameters of the network and routers like the available bandwidth, overall delay, jitter, queue lengths, number of packets dropped in the routers, etc. This information is input to the TET. Based on this measured state, the TET performs the resource and route management in the network. The TET management tasks include Bandwidth Management (LSP setup/dimensioning, LSP preemption) and Route Management (LSP routing, traffic routing), as shown in Figure 5. The TET decides the course of action, such as to create a new LSP or to vary the capacity allocated to a given LSP or to preempt a low priority LSP

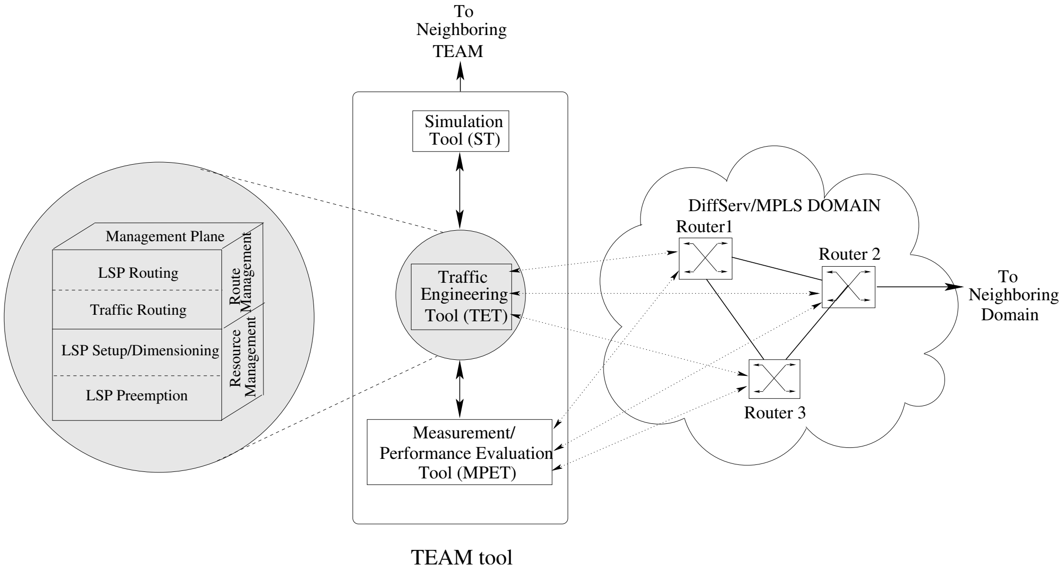


Figure 5: TEAM: Traffic Engineering Automated Manager.

to accommodate a new one, or to establish the path for an LSP requiring a specified QoS. The TET also automatically implements the action, configuring accordingly the routers and switches in the domain. Whenever required, the TET can consolidate the decision using the ST. The ST simulates a network with the current state of the managed network and applies the decision of the TET to verify the achieved performance.

Currently, TEAM is designed for the complete automated management of a single MPLS Autonomous System (AS). Research efforts are under-way to extend the operation of TEAM for multi-domain management and also manage the underlying optical network in conjunction with the MPLS network.

2.4.1 Traffic Engineering Tool (TET)

The Traffic Engineering Tool is the most important component of TEAM. This tool is responsible for the resource and route management in the network, taking the decisions related to such management tasks and implementing them in the network. The TET makes use of the two other TEAM components, MPET and ST, in order to optimize the management of the network domain.

To illustrate the inter-relations of the listed problems for MPLS network management, consider the scenario where the MPLS network has been designed initially and needs to be managed efficiently to handle the various network events. Possible network events could be arrival of a request for LSP setup based on the Service Level Specification (SLS) agreements or arrival of a bandwidth request. The first event can be handled by the combined use of the LSP routing and LSP preemption. The LSP routing aims to find the route on the physical network over which the LSP will be routed. LSP preemption decides if any existing LSPs need to be preempted on the route to make way for the new LSP if there is not enough available bandwidth. The second event of arrival of a bandwidth request triggers the traffic routing and the LSP setup and dimensioning which may in turn trigger the LSP creation steps of routing and preemption. Traffic routing decides the route to be taken by the bandwidth request in the MPLS network. The LSP setup and dimensioning procedure decides if a new direct LSP should be established or the existing direct LSP should be re-dimensioned and the capacity to be allocated in either case.

2.4.2 Measurement/Performance Evaluation Tool (MPET)

The Measurement and Performance Evaluation Tool (MPET) is used to measure the network state to be reported to the TET and also to check if the TET decisions that have been implemented have the intended effect on the network. Currently, the available bandwidth is considered as the most important state variable in the network that provides a sufficient glimpse of the network. Thus, the MPET implementation measures the available bandwidth of the network links reliably.

The available bandwidth on a link is indicative of the amount of load that can be routed on the link. Obtaining an accurate measurement of the available bandwidth can be crucial to effective deployment of QoS services in a network. Based on the available bandwidth in the network, the network manager can obtain information about the congestion in the network, decide the admission control, perform routing etc. For MPLS networks, the available bandwidth information can be used to decide about the LSP setup, LSP routing, LSP preemption etc. Each of these processes needs available bandwidth information at a suitable time-scale. It is desirable to obtain the available bandwidth information by measurements from the actual links and LSPs because they give more realistic information about the available bandwidth. The nominal available bandwidth information can be obtained by subtracting the nominal reservation for the LSPs from the link capacity which gives a lower bound. More often than not, the traffic requests do not utilize the full reservation, in which case it is beneficial to obtain accurate measurements of the network available bandwidth. The available bandwidth can be measured both for a link and for an end-to-end path.

2.4.3 Simulation Tool (ST)

The Simulation Tool (ST) is a comprehensive code which implements each of the policies in use by the TET. In order to help TEAM to take optimal decisions, the TET may use the ST to consolidate the decisions taken. The ST simulates a network with the current state of the managed network and applies the decision of the TET to verify the achieved performance. The TET management tasks that can be simulated by ST include LSP setup/dimensioning, LSP preemption, LSP re-dimensioning and LSP routing.

The goal of the Simulation Tool is to respond in real-time to the network events. The real-time responsiveness is essential for the TEAM software since the network events have to be handled when they occur. Since the execution of simulations is the most time-consuming task in an on-line simulation system, the methods suggested in [83] are used to speed up the on-line simulation. The first method is to parallelize the execution of the simulations, encapsulating each simulation in a thread and distributing the thread across machines. The second method of topology decomposition is used for speeding up the execution of a single simulation method. The simulation language C++ was chosen to implement the ST in an object-oriented manner.

ST will help TET, and therefore TEAM, to take accurate decisions in order to provide the requested QoS for each end-to-end connection.

In the following chapters of this thesis, individual algorithms and policies for TEAM components are described and their performance evaluated when they are implemented on an individual basis. The algorithms for LSP setup/dimensioning and QoS estimation based path selection for traffic routing are given in Chapter 3 and Chapter 4, respectively. These algorithms are used by the TET component of TEAM. The available bandwidth measurement algorithms for a link and end-to-end path are given in Chapter 5 and Chapter 6, respectively. These algorithms are used by the MPET component of TEAM. The extension of TEAM for inter-domain management is given in Chapter 7. The algorithms and performance evaluations for the other components of TEAM can be found in [86]. After these individual algorithms, Chapter 8 describes the implementation of TEAM on a physical testbed. The software structure of TEAM is described along with the performance evaluation for the comprehensive implementation of TEAM as a whole.

Chapter 3

Optimal Policy for LSP Setup

The bandwidth resources of a DiffServ/MPLS network need to be efficiently managed. In this chapter, a novel optimal decision policy is presented for online design of DiffServ/MPLS networks. The policy decides when to setup a new LSP, re-dimension an existing one or route the traffic on a simple hop-by-hop IP route optimally. The policy is based on a traffic-driven approach and balances bandwidth, signaling and switching costs. Furthermore, since a given traffic load may change depending on time, the policy also performs filtering in order to avoid oscillations which may occur in case of variable traffic. A greedy version of the policy was first introduced in [87], which was later optimized in [88].

This chapter is organized as follows: The motivation for the development of the optimal policy is given in Section 3.1. In Section 3.2, related work for the LSP setup policy is presented. Then, in Section 3.3, the LSP setup problem is formulated, various definitions are explained, and an illustrative example is given. Next, in Section 3.4, the new optimal LSP setup policy is formulated and obtained. A sub-optimal policy for LSP setup is then presented in Section 3.5 because the optimal policy is computationally expensive for large networks. Numerical results are analyzed in Section 3.6.

3.1 Motivation

An important aspect in designing a DiffServ/MPLS network is to determine an initial topology and to adapt it to the traffic load. A topology change in an MPLS network occurs when a new LSP is created between two nodes. The LSP creation involves determining the route of the LSP and the according resource allocation to the path. A fully connected MPLS network can be used to minimize the signaling. The objective of this chapter is to determine when an LSP should be created and how often it should be re-dimensioned.

It is necessary to define a mapping between the DiffServ classes and the LSPs in the DiffServ/MPLS network to achieve efficient resource utilization. This mapping is still an open research problem. Towards this end, Class-Types (CTs) have been defined in [24] which are then mapped to virtual MPLS networks. Each virtual MPLS network will have its own topology which will be independent of other virtual networks. This will provide better resource utilization by performing traffic engineering at DiffServ level. Also the LSPs can be mapped over a pure-MPLS (non-DiffServ) network extending DiffServ mapping to heterogeneous networks. Three class types have been defined, with each being carried on a virtual MPLS network by itself, *i.e.* ,

- *MPLS net1* as Class type 0*, i.e., Best Effort* (BE)
- *MPLS net2* as Class type 1*, i.e., Expedited Forwarding* (EF) (for real time traffic)
- *MPLS net3* as Class type 2*, i.e., Assured Forwarding 1 and 2* (AF) (for low loss classes)

These virtual networks are layered on top of the physical network, as illustrated in Figure 4 in Chapter 1. The capacity of each physical link is partitioned among different MPLS networks, and a maximum capacity (fixed percentage of the total link capacity) is assigned to each partition. The unused reserved bandwidth can then be used for Best Effort (BE) traffic. The design and management of these MPLS networks is a fundamental key to the success of the DiffServ-MPLS mapping. However, many problems such as *the definition of the network topology, LSP dimensioning, LSP set-up/tear-down procedures, LSP routing, and LSP adaptation for incoming resource requests*, need to be solved. The classical network design methods, which are performed off-line by using a-priori known traffic demand, are not suitable for MPLS networks due to the high unpredictability of the Internet traffic.

3.2 Related Work

A fully connected MPLS network, where every pair of LSRs is connected by a direct LSP, is very inefficient due to the high signaling cost and the management of a large number of LSPs [34]. The signaling cost is in the order of N^2, where N is the total number of routers.

Two different approaches, *traffic-driven* and *topology-driven*, can be used for MPLS network design. In the *traffic-driven* approach, the LSP is established on demand according to a request for a flow, traffic trunk or bandwidth reservation. The LSP is released when the request becomes inactive. In the *topology-driven* approach, the LSP is established in advance according to the routing protocol information, e.g., when a routing entry is generated by the routing protocol. The LSP is maintained as long as the corresponding routing entry exists, and it is released when the routing entry is deleted. The advantage of the traffic-driven approach is that only the required LSPs are set-up; while in the topology-driven approach, the LSPs are established in advance even if no data flow occurs.

Much of the research effort in the MPLS networks is based on the assumption that the request for the setup of an LSP is received by an ingress LSR from a Service Level Agreement (SLA) for the traffic, the LSP is manually setup and the traffic then utilizes the LSP. Very few research proposals have addressed the traffic-driven LSP setup and tear-down decision problem. One such traffic-driven LSP setup policy has been proposed in [34], in which an LSP is established whenever the number of bytes forwarded within one minute exceeds a threshold. This policy reduces the number of LSPs in the network; however, it has very high signaling costs and needs high control efforts for variable and bursty traffic as in the case of a fully connected network. Another traffic driven LSP setup method for multicast in an MPLS network is described in [35].

In [89], the authors bring up the issue that the number of LSPs for a fully connected point-to-point (p-t-p) network is $O(N^2)$, where n is the number of edge nodes. As a solution to this issue, they propose a Traffic Engineering strategy using multiple multipoint-to-point (m-t-p) LSPs, which brings the number of required LSPs to $O(N)$. In terms of graph theory, the shape of the LSP is an inverted tree rooted at an egress node. The set of m-t-p LSPs created satisfies the requirement that they provide a diversity of routes including at least two routes which do not share any single node to each individual ingress/egress node pair for effective load balance and reliability. The authors also proposed an effective global flow assignment that included fault considerations. Although the approach was shown to reduce the number of overall LSPs when compared to p-t-p approaches, and also fault recovery was introduced in the method, the effect of selecting longer paths due to the routing restrictions and the consideration of LSP preemption is difficult in an m-t-p approach.

In [90], an end-to-end setup mechanism of a Constraint-based Routed LSP (CR-LSP) initiated by the ingress LSR is proposed. The authors also specify mechanisms to provide means for reservation of resources using Label Distribution Protocol (LDP). This mechanism can be used together with the proposed LSP setup policy.

In an earlier version [87] of the proposed policy, a threshold-based policy for LSP set-up is presented. The policy is a traffic-driven approach and balances the bandwidth, signaling and switching costs. However, the policy is greedy since it minimizes only the instantaneous costs. In this thesis, the optimal version of the policy is proposed. It is shown that the optimal policy is a threshold-based policy, using the Markov Decision Process (MDP) [91] theory.

3.3 Setup Problem Formulation

Consider a physical network $G_{\text{ph}}(N, L)$ with a set of N routers and a set of physical links L. The following notation for $G_{\text{ph}}(N, L)$ is defined:

- $l(i, j) \in L$: There exists a physical link between routers i and j,
- $C_{\text{ph}}(i, j)$ for $i, j \in N$: The total link capacity of $l(i, j)$,
- $h(i, j)$ for $i, j \in N$: number of hops between the nodes i and j.

Also consider a virtual "induced" MPLS network $G(N, \mathcal{L})$, overlaying the physical network $G_{\text{ph}}(N, L)$. This virtual MPLS network consists of the same set of routers N as the physical network $G_{\text{ph}}(N, L)$, and a set of LSPs, denoted by $\mathcal{L}$. It is assumed that each link $l(i, j)$ of the physical network corresponds to a default LSP in $\mathcal{L}$ which is non-removable. The other elements of $\mathcal{L}$ are the LSPs (virtual links) built between non-adjacent nodes of $G_{\text{ph}}(N, L)$ and routed over multiple physical links. Note that G_{ph} and G are directed graphs and $\mathcal{L} \supseteq L$. In other words, the different MPLS networks (for different class-types) are built by adding virtual LSPs to the physical topology when needed. The following notations are defined for $G(N, \mathcal{L})$:

- LSP$(i, j) \in \mathcal{L}$: LSP between routers i and j (when they are not physically connected),
- LSP$_0(i, j) \in \mathcal{L}$: default LSP between routers i and j (when they are physically connected),
- $C(i, j)$ for $i, j \in N$: total capacity of LSP(i, j) ($C(i, j) = 0 \Leftrightarrow$ LSP(i, j) not established),
- $A(i, j)$ for $i, j \in N$: available capacity on LSP(i, j) ($A(i, j) = 0 \Leftrightarrow$ LSP(i, j) fully occupied),
- $B(i, j)$ for $i, j \in N$: total bandwidth reserved between routers i and j. It represents the total traffic between router i as the source and router j as the destination.

The default and non-default LSPs can be explained with the help of Figure 6. The dotted lines between nodes 1-4, 4-6, and 6-8 represent the default-LSPs and the thick line between nodes 1-8 represents the non-default direct LSP which is routed over the default-LSPs.

All the default LSPs in $G_{\text{ph}}(N, L)$ are assumed to have large capacity which is available to be borrowed by the other multi-hop LSPs that will be routed over the corresponding physical links $l(i, j)$. Each non-default LSP must be routed on a shortest path in $G_{\text{ph}}(N, L)$.

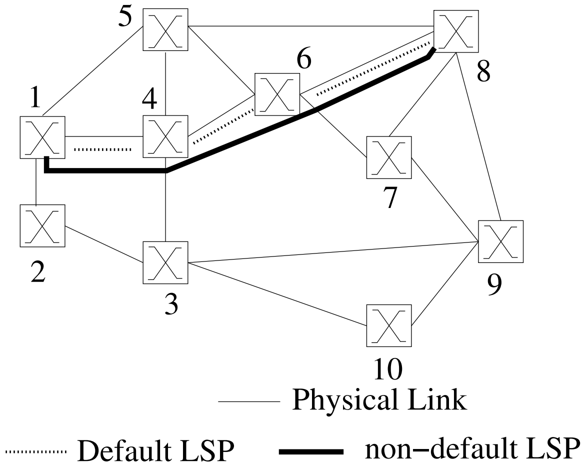


Figure 6: Default and Non-default LSPs.

The shortest path $P_{\text{ph}}(i, j)$ between a source node i and destination node j is the minimum hop path in $G_{\text{ph}}(N, L)$ and is denoted by:

$$P_{\text{ph}}(i, j) = \{l(i, h), \ldots, l(k, j)\}$$

In the MPLS network, the bandwidth requests between i and j are routed either on a direct LSP(i, j) or on $P(i, j)$, which is a multiple-LSP path overlaying $P_{\text{ph}}(i, j)$:

$$P(i, j) = \{\text{LSP}(i, h), \ldots, \text{LSP}(k, j)\}$$

The physical links are assumed to have sufficiently large $C_{\text{ph}}(i, j)$ so that whenever any LSP is re-dimensioned, it can borrow bandwidth from the physical links that it passes through. The following two quantities can be defined:

- $B_L(i, j)$ for $i, j \in N$: part of $B(i, j)$ that is routed over LSP(i, j),

- $B_P(i, j)$ for $i, j \in N$: part of $B(i, j)$ that is routed over $P(i, j)$.

Note that $B(i, j) = B_L(i, j) + B_P(i, j)$ is the total bandwidth requests between i and j, $C(i, j) = A(i, j) + B_L(i, j)$ is the total capacity of LSP(i, j) and $B_P(i, j) = 0$ for default-LSPs since $P(i, j)$ coincides with the LSP$_0(i, j)$.

Let $S(i,j)$ be the set of all LSP(u,v) such that the corresponding shortest path $P_{\text{ph}}(u,v)$ contains the link $l(i,j)$. The following condition must be satisfied:

$$\sum_{\text{LSP}(u,v)\in S(i,j)} C(u,v) \leq \delta C_{\text{ph}}(i,j) \tag{1}$$

where $\delta < 1$ is a maximum fraction of $C_{\text{ph}}(i,j)$ that can be assigned to LSPs. This means that the sum of capacity of all LSPs using a particular physical link on their path must not exceed a portion δ of the capacity of that physical link.

3.3.1 Definitions

The following definitions will be used in the LSP setup problem formulation:

• *Definition 1: Decision instants and bandwidth requests*

Let t_m be the arrival instant of a new bandwidth request between routers i and j for the amount $b(i,j)$. The instant t_m is called a *decision instant* because a decision has to be made to accommodate the arrival of the new bandwidth request. When a new bandwidth request $b(i,j)$ arrives in the MPLS network at instant t_m, the existence of a direct LSP between i and j is checked initially. For direct LSP between i and j, the available capacity $A(i,j)$ is then compared with the request $b(i,j)$. If $A(i,j) > b(i,j)$, then the requested bandwidth is allocated on that LSP and the available capacity is reduced accordingly. Otherwise, $C(i,j)$ can be increased subject to condition 1 in order to satisfy the bandwidth request. If no direct LSP exists between i and j, then a decision needs to be made whether to setup a new LSP and its according $C(i,j)$. Every time a new LSP is setup or re-dimensioned, the previously granted bandwidth allocation requests between i and j routed on $P(i,j)$ are re-routed on the new direct LSP(i,j). However, this rerouting operation is only virtual, since both LSP(i,j) and $P(i,j)$ are routed on the physical network over the same $P_{\text{ph}}(i,j)$.

Let t_n be the departure instant of a request for bandwidth allocation $b(i,j)$ routed on LSP(i,j). At this instant, a decision needs to be made whether or not to re-dimension LSP(i,j), *i.e.* , reduce its capacity $C(i,j)$.

It is assumed that the events and costs associated with any given node pair i and j are independent of any other node pair. This assumption is based on the fact that the new bandwidth requests are routed either on the direct LSP between the source and destination or on $P(i, j)$, *i.e.* , the other LSPs are not utilized for routing the new request. This assumption allows the analysis to be carried for any node pair and be guaranteed that it will be true for all other pairs. Under this assumption, the explicit dependence of the notations on i and j can be dropped. Also, it is assumed that nodes i and j are not physically connected. For the default LSPs, there is a large amount of available bandwidth and they too borrow bandwidth, in large amounts, from the physical links, if needed.

- *Definition 2: Set of events*

For each router pair i and j in the MPLS network, e_m is the *event* observed at t_m.

- $e_m = 1$ if there is an arrival of a bandwidth request for amount b,
- $e_m = 0$ if there is a departure of a request of amount b from $B_P(i, j)$,
- $e_m = 2$ if there is a departure of a request b from $B_L(i, j)$.

- *Definition 3: Set of states*

For each router pair i and j in the MPLS network, the state vector s_m at a given time-instant t_m, $m = 0, 1, ...$ is defined as:

$$s_m(i, j) = [A, B_L, B_P] \tag{2}$$

where A is the available capacity on LSP(i, j), B_L is the part of B that is routed over LSP(i, j) and B_P is the part of B routed on $P(i, j)$. Note that the state space $\overline{s}$, the set of all system states, is finite since A is limited by C which is in turn limited by the minimum of the link bandwidths on P_{ph}. B_L is limited by C and B_P by the minimum of default LSP bandwidths on P_{ph}. Also note that states with non-zero $A(i, j)$ and $B_P(i, j)$ are possible because just before the instant of observation, some user request might have departed leaving available bandwidth in LSP(i, j). The state information for each LSP is stored in the first router of the LSP.

• *Definition 4: Set of extended states*

The state space $\overline{s}$ of the system can be extended by the coupling of the current state and the event.

$$S_m = \langle s_m, e_m \rangle \, . \tag{3}$$

The set $\overline{S}$ of extended states S_m is the basis for determining the decisions to be taken to handle the events.

• *Definition 5: Set of actions*

The decision of setting up or re-dimensioning LSP(i, j) when the event e_m occurs, is captured by the binary action variable $a \in A = \{0, 1\}$.

- $a = 1$ means that LSP(i, j) will be setup or re-dimensioned and the new value of its capacity is set as $C = B_L + B_P + b$, where b is considered negative if the event is a departure, either over LSP(i, j) or $P(i, j)$,
- $a = 0$ means that no action will be taken on the capacity of LSP(i, j).

• *Definition 6: Decision rules and policies*

A decision rule d_i provides an action selection in each state at a given decision instant t_i and a policy π specifies the decision rules to be used in each decision instant, *i.e.* $\pi = \{d_0(\overline{S}), d_1(\overline{S}), d_2(\overline{S}), \ldots\}$. If $d_i(\overline{S}) = d_j(\overline{S}) \forall i$ and j, then the policy is stationary as the decision is independent of the time instant. For most of the possible system states, the decision rule chooses an action from the set $\{0, 1\}$ but there are a few states where only one action is possible. Those states and corresponding actions are:

- $S_m = \langle [A, B_L, B_P], 1\rangle$ where $A > b \Rightarrow a = 0$ (the new request routed on LSP(i, j))

- $S_m = \langle [A, B_L, B_P], 0\rangle \Rightarrow a = 0$ (the request ending on $P(i, j)$)

- $S_m = \langle [A, B_L, 0], 2\rangle$ where $B_L = b \Rightarrow a = 1$ (LSP(i, j) is torn down)

• *Definition 7: Cost function*

The incremental cost $W(S, a)$ for the system in state s, occurrence of the event e, and the taken action a is

$$W(S, a) = W_{\text{sign}}(S, a) + W_{\text{b}}(S, a) + W_{\text{sw}}(S, a) \tag{4}$$

where $W_{\text{sign}}(S, a)$ is the cost for signaling the set-up or re-dimensioning of the LSP to the involved routers, $W_{\text{b}}(S, a)$ is the cost for the carried bandwidth and $W_{\text{sw}}(S, a)$ is the cost for switching of the traffic. The cost components depend on the system state and the action taken for an event.

The signaling cost $W_{\text{sign}}(S, a)$ is incurred instantaneously only when action $a = 1$ is chosen for state S. It accounts for the signaling involved in the process of setup or re-dimensioning of the LSP. This cost depends linearly on the number of hops h in $P_{\text{ph}}(i, j)$ over which the LSP is routed, plus a constant component to take into account the notification of the new capacity of the LSP to the network. Thus

$$W_{\text{sign}}(S, a) = a[c_{\text{s}}h + c_{\text{a}}] \tag{5}$$

where c_{s} is the coefficient for signaling cost per hop and c_{a} is the fixed notification cost coefficient. This cost is not incurred if $a = 0$.

The other two components of Eq. 4 relate to the bandwidth (w_{b}) and switching (w_{sw}) cost rates, respectively.

$$\begin{aligned} W_{\text{b}}(S, a) &= \int_0^T w_{\text{b}}(S, a)\, dt \\ W_{\text{sw}}(S, a) &= \int_0^T w_{\text{sw}}(S, a)\, dt \end{aligned}$$

where T is the time till the next event, *i.e.*, until the system stays in state S.

The bandwidth cost rate $w_{\text{b}}(S,a)$ to reserve $(B_L + B_P)$ capacity units depends linearly on $(B_L + B_P)$ and on the number of hops $h(i,j)$ in the physical shortest path over which the request is routed.

$$w_{\text{b}}(S,a) = c_{\text{b}} h(B_L + B_P), \tag{6}$$

where c_{b} is the bandwidth cost coefficient per capacity unit (c.u.) per time. Note that, from the routing assumption, the physical path is the same for LSP(i,j) and for $P(i,j)$ and thus the bandwidth cost rate depends only on the total carried bandwidth, irrespective of the fractions carried over different paths.

The switching cost rate $w_{\text{sw}}(S,a)$ depends linearly on the number of switching operations in IP or MPLS mode and the switched bandwidth. The total number of switching operations is always h since the physical path is fixed. Whether these switching operations are IP or MPLS depends on the path chosen in the MPLS network. For B_L c.u. routed on LSP(i,j), 1 router performs IP switching and $(h-1)$ routers perform MPLS switching. For B_P c.u. routed on $P(i,j)$, h routers perform IP switching.

$$w_{\text{sw}}(S,a) = [c_{\text{ip}} + c_{\text{mpls}}(h-1)]B_L + hc_{\text{ip}}B_P, \tag{7}$$

where c_{ip} and c_{mpls} are the switching cost coefficients per c.u. per time in IP and MPLS mode respectively. Summarizing, the signaling cost is incurred only at decision instants when $a = 1$, while the bandwidth and switching costs are accumulated continuously until a new event occurs.

3.3.2 Example

This example illustrates how the state vectors defined in Eq. 2 are varied by bandwidth request arrival and LSP setup. Consider a simple three node network where node 1 is connected to node 2 and node 2 to node 3. Suppose, at the initial instant t_0, the state vectors for the three nodes are given as follows (capacity is expressed in capacity units):

$$s_0(1,2) = (1000, 14, 0); s_0(2,3) = (1000, 15, 0); s_0(1,3) = (0, 0, 5).$$

Suppose that two alternative events occur at instant t_0 and time till the next event is T:

EVENT A: A bandwidth request for 2 c.u. arrives between nodes 1 and 2. Then $B_L(1,2)$ increases to 16 and $A(1,2)$ reduces by 2. So the new state vectors become:

$$s_A(1,2) = (998, 16, 0); s_A(2,3) = (1000, 15, 0); s_A(1,3) = (0, 0, 5).$$

EVENT B: A bandwidth request for 10 c.u. arrives between nodes 1 and 3.

If the decision process results in $a = 0$, the request is routed on the 2-LSP path $P(1,3)$ and the new state vectors are

$$s_{B_0}(1,2) = (985, 29, 0); s_{B_0}(2,3) = (985, 30, 0); s_{B_0}(1,3) = (0, 0, 15).$$

The incremental cost from initial state $s_0(1,3)$ is calculated from Eq. 4 as:

$$\begin{aligned} W_1(S,a) &= W_{\text{b}}(S,a) + W_{\text{sw}}(S,a) + W_{\text{sign}}(S,a) \\ &= \{c_{\text{b}} \cdot 2 \cdot 15 + (c_{\text{ip}} \cdot 2) \cdot 15\}T + 0. \end{aligned}$$

If the decision process determines $a = 1$, a direct LSP between nodes 1 and 3 is created and the new state vectors are

$$s_{B_1}(1,2) = (985, 14, 0); s_{B_1}(2,3) = (985, 15, 0); s_{B_1}(1,3) = (0, 15, 0).$$

The incremental cost from initial state $s_0(1,3)$ is calculated from Eq. 4 as:

$$\begin{aligned} W_1(S,a) &= W_{\text{b}}(S,a) + W_{\text{sw}}(S,a) + W_{\text{sign}}(S,a) \\ &= \{c_{\text{b}} \cdot 2 \cdot 15 + (c_{\text{ip}} + c_{\text{mpls}}) \cdot 15\}T + 2c_{\text{s}} + c_{\text{a}}. \end{aligned}$$

The set of all possible system states (Definition 4), events (Definition 2), actions (Definition 5) and associated costs (Definition 7) is given in Table 1. In the table, the node pair (i,j) is implicit and T is the time interval between current event and the next event. In all the cost formulations, the first component refers to the cost incurred for the bandwidth carried, second component refers to the cost for switching of the traffic over LSP(i,j) or $P(i,j)$, and third, if it exists, to the signaling cost.

Table 1: Set of possible states.

Old state	Action	New state	Cost
$\langle [A, B_L, B_P], 0\rangle$	0	$\langle [A, B_L, B_P - b], e\rangle$ where $e \in \{0, 1, 2\}$	$T\left[hc_{\text{b}}\{B_L + B_P - b\}\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} B_L$ $+ hc_{\text{ip}}\,(B_P - b)\right]$
$\langle [A, B_L, B_P], 1\rangle$ where $A \geq b$	0	$\langle [A - b, B_L + b, B_P], e\rangle$ where $e \in \{0, 1, 2\}$	$T\left[hc_{\text{b}}\{B_L + B_P + b\}\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} (B_L + b)$ $+ hc_{\text{ip}} B_P\right]$
$\langle [A, B_L, B_P], 1\rangle$ where $A < b$	0	$\langle [A, B_L, B_P + b], e\rangle$ where $e \in \{0, 1, 2\}$	$T\left[hc_{\text{b}}\{B_L + B_P + b\}\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} B_L$ $+ hc_{\text{ip}}\,(B_P + b)\right]$
$\langle [A, B_L, B_P], 1\rangle$ where $A < b$ and $Y = B_L + B_P + b$	1	$\langle [0, Y, 0], e\rangle$ where $e \in \{1, 2\}$	$T\left[hc_{\text{b}} Y\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} Y\right]$ $+ (c_{\text{s}} h + c_{\text{a}})$
$\langle [A, B_L, B_P], 2\rangle$	0	$\langle [A + b, B_L - b, B_P], e\rangle$ where $e \in \{0, 1, 2\}$	$T\left[hc_{\text{b}}\{B_L + B_P - b\}\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} (B_L - b)$ $+ hc_{\text{ip}} B_P\right]$
$\langle [A, B_L, B_P], 2\rangle$ where $Y = B_L + B_P - b$	1	$\langle [0, Y, 0], e\rangle$ where $e \in \{1, 2\}$	$T\left[hc_{\text{b}} Y\right]$ $+T\left[\left\{c_{\text{ip}} + (h-1)\, c_{\text{mpls}}\right\} Y\right]$ $+ (c_{\text{s}} h + c_{\text{a}})$

3.4 Optimal Setup Policy

A stochastic model is used to determine the optimal decision policy for LSP setup. The optimization problem is formulated as a Continuous-Time Markov Decision Process (CT-MDP) [91].

3.4.1 Optimization Problem Formulation

The cost functions for the MDP theory have been defined in Definition 7. Following the theory of MDPs, the expected infinite-horizon discounted total cost, $v^{\pi}(S_0)$, with discounting rate α, given that the process occupies state S_0 at the first decision instant and the decision policy is π is given by:

$$v^{\pi}_{\alpha}(S_0) = E^{\pi}_{S_0}\left\{\sum_{m=0}^{\infty} e^{-\alpha t_m}\left[W_{\text{sign}}(S_m, a) + \int_{t_m}^{t_{m+1}} e^{-\alpha(t - t_m)}[w_{\text{b}}(S_m, a) + w_{\text{sw}}(S_m, a)]dt\right]\right\}. \tag{8}$$

where $t_0, t_1, \ldots$ represent the times of successive events and $W_{\text{sign}}(S_m, a)$ represents the fixed part of the cost incurred whereas $[w_{\text{b}}(S_m, a) + w_{\text{sw}}(S_m, a)]$ represents the continuous part of the cost between times t_m and t_{m+1}.

The *optimization objective* is to find a policy π^* such that:

$$v_\alpha^{\pi^*}(s) = \inf_{\pi \in \Pi} v_\alpha^\pi(s).$$

The optimal decision policy can be found by solving the optimality equations [91] for each initial state S. The bandwidth requests are assumed to arrive according to a Poisson process with rate λ and the request durations are exponentially distributed with rate μ. With the assumptions of a discounted infinite-horizon CTMDP, the optimality equations can be written as:

$$v(S) = \min_{a \in A} \left\{ r(S, a) + \frac{\lambda + \mu}{\lambda + \mu + \alpha} \sum_{j \in \overline{S}} q(j \mid S, a) v(j) \right\}, \tag{9}$$

where $r(S, a)$ is the expected discounted cost between two decision instants and $q(j \mid S, a)$ is the probability that the system occupies state j at the subsequent decision instant, given that the system is in state S at the earlier decision instant and action a is chosen. From Eq. 4, $r(S, a)$ can be written as

$$r(S, a) = W_{\text{sign}}(S, a) + E_S^a \left\{ \int_0^{\tau_1} e^{-\alpha t} [w_{\text{b}}(S, a) + w_{\text{sw}}(S, a)] dt \right\}, \tag{10}$$

where E_S^a represents the expectation with respect to the request duration distribution and τ_1 represents the time before the next event occurs.

With the markovian assumption on request arrival and duration, the time between any two successive events (arrival of requests or departure of a request) is exponentially distributed with rate $(\lambda + \mu)$. Recalling that between two successive events, the state of the system does not change, Eq. 10 can be rewritten as follows:

$$\begin{aligned} r(S, a) &= W_{\text{sign}}(S, a) + [w_{\text{b}}(S, a) + w_{\text{sw}}(S, a)] E_S^a \left\{ \int_0^{\tau_1} e^{-\alpha t} dt \right\} \\ &= W_{\text{sign}}(S, a) + [w_{\text{b}}(S, a) + w_{\text{sw}}(S, a)] / (\alpha + \lambda + \mu). \end{aligned}$$

Table 2: Transition probabilities $q(j \mid S, a)$.

Probability	S	a	j
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 0\rangle$	0	$\langle A, B_L, B_P - b, 1\rangle$
P_{D}	$\langle A, B_L, B_P, 0\rangle$	0	$\langle A, B_L, B_P - b, 0\rangle$
$(\mu/(\lambda+\mu)) - P_{\text{D}}$	$\langle A, B_L, B_P, 0\rangle$	0	$\langle A, B_L, B_P - b, 2\rangle$
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A - b, B_L + b, B_P, 1\rangle\ \ (A \geq b)$
P_{D}	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A - b, B_L + b, B_P, 0\rangle\ \ (A \geq b)$
$(\mu/(\lambda+\mu)) - P_{\text{D}}$	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A - b, B_L + b, B_P, 2\rangle\ \ (A \geq b)$
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A, B_L, B_P + b, 1\rangle\ \ (A < b)$
P_{D}	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A, B_L, B_P + b, 0\rangle\ \ (A < b)$
$(\mu/(\lambda+\mu)) - P_{\text{D}}$	$\langle A, B_L, B_P, 1\rangle$	0	$\langle A, B_L, B_P + b, 2\rangle\ \ (A < b)$
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 1\rangle$	1	$\langle 0, B_L + B_P + b, 0, 1\rangle\ \ (A < b)$
$(\mu/(\lambda+\mu))$	$\langle A, B_L, B_P, 1\rangle$	1	$\langle 0, B_L + B_P + b, 0, 1\rangle\ \ (A < b)$
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 2\rangle$	0	$\langle A + b, B_L - b, B_P, 1\rangle$
P_{D}	$\langle A, B_L, B_P, 2\rangle$	0	$\langle A + b, B_L - b, B_P, 0\rangle$
$(\mu/(\lambda+\mu)) - P_{\text{D}}$	$\langle A, B_L, B_P, 2\rangle$	0	$\langle A + b, B_L - b, B_P, 2\rangle$
$(\lambda/(\lambda+\mu))$	$\langle A, B_L, B_P, 2\rangle$	1	$\langle 0, B_L + B_P - b, 0, 1\rangle$
$(\mu/(\lambda+\mu))$	$\langle A, B_L, B_P, 2\rangle$	1	$\langle 0, B_L + B_P - b, 0, 2\rangle$

Since the set of possible actions A is finite and $r(S, a)$ is bounded, it can be proved that the optimal policy π^* is stationary and deterministic [91].

3.4.2 Transition Probability Function

The transition probabilities $q(j\,|s, a)$ in Eq. 9 for the model are related to the transition probabilities in a M/M/1 queue and are given by Table 2. In the table, P_{D} is the probability that a connection that is departing was routed over $P(i, j)$. The probability is 0 for the states not mentioned in the table.

3.4.3 Optimality Equations

The optimality equation (Eq. 9) can be explicitly written for all possible states by obtaining $r(S, a)$ from Eq. 11 and $q(j \mid S, a)$ from Table 2 as follows:

$$v\left(\langle A, B_L, B_P, 0\rangle\right) = \frac{h\, c_{\text{b}}(B_L + B_P - b) + h\, c_{\text{ip}}\,(B_P - b) + D\, B_L}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} J, \tag{11}$$

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 1\rangle\right) = \\
&\qquad \frac{h\, c_{\text{b}}(B_L + B_P + b) + h c_{\text{ip}}\, B_P + D\ (B_L + b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} K \quad \text{for A} \geq b, \qquad (12)\\
&v\left(\langle A, B_L, B_P, 1\rangle\right) = \\
&\qquad \min\left\{ \frac{h\, c_{\text{b}}(B_L + B_P + b) + h c_{\text{ip}}\,(B_P + b) + D\, B_L}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} L, \right.\\
&\qquad \left. c_{\text{s}}\, h + c_{\text{a}} + \frac{h\, c_{\text{b}}(B_L + B_P + b) + D\ \{B_L + B_P + b\}}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} M, \right\} \qquad (13)\\
&\qquad\qquad\qquad \text{for A} < b,
\end{aligned}
$$

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 2\rangle\right) = \\
&\qquad \min\left\{ \frac{h\, c_{\text{b}}(B_L + B_P - b) + h\, c_{\text{ip}}\, B_P + D\ (B_L - b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} X, \right.\\
&\qquad \left. c_{\text{s}}\, h + c_{\text{a}} + \frac{h\, c_{\text{b}}(B_L + B_P - b) + D\ (B_L + B_P - b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} Y \right\}, \qquad (14)
\end{aligned}
$$

where

$$
\begin{aligned}
D &= \left\{ c_{\text{ip}} + (h-1) c_{\text{mpls}} \right\}, \\
J &= \left[\frac{\lambda}{\lambda + \mu} v\left(\langle A, B_L, B_P - b, 1\rangle\right) + P_{\text{D}} v\left(\langle A, B_L, B_P - b, 0\rangle\right) \right.\\
&\qquad \left. + \left\{ \frac{\mu}{\lambda + \mu} - P_{\text{D}} \right\} v\left(\langle A, B_L, B_P - b, 2\rangle\right) \right], \\
K &= \left[\frac{\lambda}{\lambda + \mu} v\left(\langle A - b, B_L + b, B_P, 1\rangle\right) + P_{\text{D}} v\left(\langle A - b, B_L + b, B_P, 0\rangle\right) \right.\\
&\qquad \left. + \left\{ \frac{\mu}{\lambda + \mu} - P_{\text{D}} \right\} v\left(\langle A - b, B_L + b, B_P, 2\rangle\right) \right], \\
L &= \left[\frac{\lambda}{\lambda + \mu} v\left(\langle A, B_L, B_P + b, 1\rangle\right) + P_{\text{D}} v\left(\langle A, B_L, B_P + b, 0\rangle\right) \right.\\
&\qquad \left. + \left\{ \frac{\mu}{\lambda + \mu} - P_{\text{D}} \right\} v\left(\langle A, B_L, B_P + b, 2\rangle\right) \right], \\
M &= \left[\frac{\lambda}{\lambda + \mu} v\left(\langle 0, B_L + B_P + b, 0, 1\rangle\right) + \frac{\mu}{\lambda + \mu} v\left(\langle 0, B_L + B_P + b, 0, 2\rangle\right) \right], \\
X &= \left[\frac{\lambda}{\lambda + \mu} v_1\left(\langle A + b, B_L - b, B_P, 1\rangle\right) + P_{\text{D}} v\left(\langle A + b, B_L - b, B_P, 0\rangle\right) \right.\\
&\qquad + \left\{ \frac{\mu}{\lambda + \mu} - P_{\text{D}} \right\} v\left(\langle A + b, B_L - b, B_P, 2\rangle\right), \\
Y &= \left[\frac{\lambda}{\lambda + \mu} v_2\left(\langle 0, B_L + B_P - b, 0, 1\rangle\right) + \frac{\mu}{\lambda + \mu} v\left(\langle 0, B_L + B_P - b, 0, 2\rangle\right) \right].
\end{aligned}
$$

By substituting Eq. 11 into Eq. 12, Eq. 13 and Eq. 14, the optimality equations are simplified as given below.

Optimality equations:

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 0\rangle\right) = \\
&\qquad \frac{h\, c_{\text{b}}(B_L + B_P - b) + h\, c_{\text{ip}}\,(B_P - b) + D\, B_L}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} J,
\end{aligned} \tag{15}
$$

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 1\rangle\right) = \\
&\qquad \frac{h\, c_{\text{b}}(B_L + B_P + b) + h\, c_{\text{ip}}\, B_P + D\ (B_L + b)}{\alpha + \lambda + \mu} + \frac{\lambda + \mu}{\alpha + \lambda + \mu} K \ \text{ for } A \geq b,
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 1\rangle\right) = \\
&\qquad \min\left\{v\left(\langle A, B_L, B_P{+}2b, 0\rangle\right), c_{\text{s}}h{+}c_{\text{a}} + v\left(\langle 0, B_L{+}B_P{+}b, b, 0\rangle\right)\right\} \text{ for } A < b,
\end{aligned} \tag{17}
$$

$$
\begin{aligned}
&v\left(\langle A, B_L, B_P, 2\rangle\right) = \\
&\qquad \min\left\{v\left(\langle A + b, B_L - b, B_P + b, 0\rangle\right), c_{\text{s}}! + c_{\text{a}} + v\left(\langle 0, B_L + B_P - b, b, 0\rangle\right)\right\},
\end{aligned} \tag{18}
$$

3.4.4 The Optimal Policy

The solutions of the optimality equations give the optimal values $v^*(A, B_L, B_P, e)$ of expected infinite-horizon discounted total costs. From the optimality equation (Eq. 17), it can be derived that for the state $S = \langle A, B_L, B_P, 1\rangle$ where $A < b$, the action would be

$$
a^*\langle A, B_L, B_P, 1\rangle = \begin{cases} 1 & c_{\text{s}}h + c_{\text{a}} < v^*\left(\langle A, B_L, B_P + 2b, 0\rangle\right) \\ & \qquad -v^*\left(\langle 0, B_L + B_P + b, b, 0\rangle\right), \\ 0 & \text{otherwise,} \end{cases} \tag{19}
$$

and for state $S = \langle A, B_L, B_P, 2\rangle$ the optimal action from optimality equation (Eq. 18),

$$
a^*\langle A, B_L, B_P, 2\rangle = \begin{cases} 1 & c_{\text{s}}\, h + c_{\text{a}} < v^*\left(\langle A + b, B_L - b, B_P + b, 0\rangle\right) \\ & \qquad -v^*\left(\langle 0, B_L + B_P - b, b, 0\rangle\right), \\ 0 & \text{otherwise.} \end{cases} \tag{20}
$$

This policy will be optimal if thresholds $v^*\left(\langle A, B_L, B_P{+}2b, 0\rangle\right) – v^*\left(\langle 0, B_L{+}B_P{+}b, b, 0\rangle\right)$ and $v^*\left(\langle A{+}b, B_L{-}b, B_P{+}b, 0\rangle\right) - v^*\left(\langle 0, B_L{+}B_P{-}b, b, 0\rangle\right)$ are monotone non-increasing

```
Algorithm:
1. Set $v^0(S) = 0$ for each state $S \in (\bar{S})$. Specify $\varepsilon > 0$ and set $n = 0$.
2. For each $S \in (\bar{S})$, compute $v^{n+1}(S)$ by substituting $v^n(S)$ on the right hand side of Eqs. (15), (16), (17) and (18).
3. If $\| v^{n+1} - v^n \| < \varepsilon$, go to step 4. Otherwise, increment $n$ by 1 and go to step 2.
4. For each $S \in (\bar{S})$, set $v^*(S) = v^n(S)$ and calculate the actions by substituting $v^*(S)$ into Eqs. (19) and (20). Stop.
```

Figure 7: The Value Iteration Algorithm.

which is true and can be proved through induction [91] by utilizing the linearity characteristics of the cost functions. These decisions have a control-limit structure. The values of $v^*(A, B_L, B_P, e)$ can be found by using either value iteration or policy iteration algorithm which are numerical procedures. The value iteration algorithm is shown in Figure 7.

The optimal policy $\pi^* = \{d^*, d^*, d^*, \ldots\}$ is stationary implying same decision rule at each decision instant and the decision rule is given by:

$$d^* = \begin{cases} 0 & S = \langle A, B_L, B_P, 0 \rangle \\ 0 & S = \langle A, B_L, B_P, 1 \rangle \ \text{ for } A \geq b \\ a^* \langle A, B_L, B_P, 1 \rangle & S = \langle A, B_L, B_P, 1 \rangle \ \text{ for } A < b \\ a^* \langle A, B_L, B_P, 2 \rangle & S = \langle A, B_L, B_P, 2 \rangle \end{cases} \tag{21}$$

where $a^* \langle A, B_L, B_P, 1 \rangle$ and $a^* \langle A, B_L, B_P, 2 \rangle$ are given by Eq. 19 and Eq. 20, respectively.

The threshold structure of the optimal policy facilitates the solution of the optimality equations (Eq. 15-Eq. 18) but still it is difficult to pre-calculate and store the solution because of the large number of possible system states. So, a sub-optimal policy, called the *Least One-Step Cost Policy*, that is easy and fast to calculate is proposed.

3.5 Sub-optimal Setup Policy

The proposed Least One-Step Cost policy is an approximation to the solution of the optimality equations (Eq. 15-Eq. 18). It minimizes the cost incurred between two decision instants. Instead of going through all the iterations of the value iteration algorithm, if only the first iteration is performed with the assumption that $v^0\left(\langle A, B_L, B_P-b, 0\rangle\right)=0$, then

$$\begin{aligned}
&v^1\left(\langle A, B_L, B_P, 0\rangle\right)= \\
&\quad\frac{h\,c_{\mathrm{b}}(B_L+B_P-b)+h\,c_{\mathrm{ip}}\,(B_P-b)+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}B_L}{\alpha+\lambda+\mu}, \\
&v^1\left(\langle A, B_L, B_P, 1\rangle\right)= \\
&\quad\frac{h\,c_{\mathrm{b}}(B_L+B_P+b)+h\,c_{\mathrm{ip}}\,B_P+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}(B_L+b)}{\alpha+\lambda+\mu}\quad\text{for } A\geq b, \\
&v^1\left(\langle A, B_L, B_P, 1\rangle\right)= \\
&\quad\min\left\{\frac{hc_{\mathrm{b}}(B_L+B_P+b)+hc_{\mathrm{ip}}\,(B_P+b)+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}B_L}{\alpha+\lambda+\mu},\right. \\
&\quad\left.c_{\mathrm{s}}h+c_{\mathrm{a}}+\frac{hc_{\mathrm{b}}(B_L+B_P+b)+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}\{B_L+B_P+b\}}{\alpha+\lambda+\mu}\right\}\quad\text{for } A<b, \\
&v^1\left(\langle A, B_L, B_P, 2\rangle\right)= \\
&\quad\min\left\{\frac{h\,c_{\mathrm{b}}(B_L+B_P-b)+h\,c_{\mathrm{ip}}\,B_P+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}\{B_L-b\}}{\alpha+\lambda+\mu},\right. \\
&\quad\left.c_{\mathrm{s}}\,h+c_{\mathrm{a}}+\frac{h\,c_{\mathrm{b}}(B_L+B_P-b)+\left\{c_{\mathrm{ip}}+(h-1)c_{\mathrm{mpls}}\right\}\{B_L+B_P-b\}}{\alpha+\lambda+\mu}\right\}
\end{aligned}$$

From these single-step cost formulations, the action decision can be obtained. For the state $\langle A, B_L, B_P, 1\rangle$,

$$a^1\left(\langle A, B_L, B_P, 1\rangle\right)=\begin{cases}1 & B_P+b>B_{Th},\\ 0 & \text{otherwise}\end{cases}\quad\text{for } A<b \tag{22}$$

upon comparison of the two terms of $v^1\left(\langle A, B_L, B_P, 1\rangle\right)$. Similarly, comparing the two terms of $v^1\left(\langle A, B_L, B_P, 2\rangle\right)$,

$$a^1\left(\langle A, B_L, B_P, 2\rangle\right)=\begin{cases}1 & B_P>B_{Th},\\ 0 & \text{otherwise.}\end{cases} \tag{23}$$

In both Eq. 22 and Eq. 23,

$$B_{Th}=\frac{\{c_{\mathrm{s}}\,h+c_{\mathrm{a}}\}\,\{\alpha+\lambda+\mu\}}{(h-1)\,(c_{\mathrm{ip}}-c_{\mathrm{mpls}})} \tag{24}$$

By calculating $v^1(S)$ for all $S \in \overline{S}$, the one-step cost of the infinite-horizon model is minimized. Since $v^n(S)$ in the value iteration algorithm converges to $v^*(S)$, the one-step value $v^1(S)$ is a significant part of $v^*(S)$ and is very easy to calculate.

Thus, the one-step least-cost policy $\pi^\# = \{d^\#, d^\#, d^\#, \ldots\}$ is stationary implying same decision rule at each decision instant and the decision rule is given by

$$d^\# = \begin{cases} 0 & S = \langle A, B_L, B_P, 0\rangle \\ 0 & S = \langle A, B_L, B_P, 1\rangle \text{ for } A \geq b \\ a^1 \langle A, B_L, B_P, 1\rangle & S = \langle A, B_L, B_P, 1\rangle \text{ for } A > b \\ a^1 \langle A, B_L, B_P, 2\rangle & S = \langle A, B_L, B_P, 2\rangle \end{cases} \tag{25}$$

where $a^1 \langle A, B_L, B_P, 1\rangle$ and $a^1 \langle A, B_L, B_P, 2\rangle$ are given by Eqs. 22 and 23, respectively.

The algorithm in Figure 8 can be implemented for the threshold-based sub-optimal *Least One-Step Cost Policy* for LSP set-up/re-dimensioning.

3.6 Performance Evaluation

Having identified the different parameters involved in the LSP setup policy, steps for implementing the policy are now explained. For each LSP, during its connection setup phase, the network controller assigns the cost functions based on the signaling load of the network. In the model, the cost functions are assumed to be linear ($W_{\text{sign}}, W_{\text{b}}, W_{\text{sw}}$ from Eq. 4) with respect to the bandwidth requirements of the requests. By keeping a history of user requests, the average inter-arrival time and connection duration can be estimated.

Given the input parameters (cost functions and various distributions), the value iteration algorithm (Figure 7) can be used to determine the optimal policy with the decision rule (Eq. 21). The optimal policy is the stored in a tabular format. Each entry of the table specifies the optimal decision for the possible events for all possible node pairs of the network. Whenever there is a bandwidth request arrival or departure, the network performs a table lookup at the corresponding node pair entry. Setup/re-dimensioning of the LSP is performed if the traffic not utilizing the LSP exceeds a threshold (Eq. 19 and Eq. 20). The

```
At time $t_m$, $\boldsymbol{s_m = [A, B_L, B_P]}$ and event $\boldsymbol{e}$ occurs

Case 0: $\boldsymbol{e =}$ Departure of $\boldsymbol{b}$ from $\boldsymbol{P}$
      Do not re-dimension LSP.            $s_{m+1} = [A, B_L, B_P - b]$

Case 1: $\boldsymbol{e =}$ Arrival of $\boldsymbol{b}$
      If LSP exists and $A > b$
        Do not re-dimension LSP.          $s_{m+1} = [A - b, B_L + b, B_P]$
        Request accepted and routed on LSP
      Else
        If $B_P + b > B_{Th}$
          Set-up/re-dimension LSP.        $s_{m+1} = [0, B_L + B_P + b, 0]$
          Request accepted and routed on LSP
        Else
          Do not re-dimension LSP.        $s_{m+1} = [A, B_L, B_P + b, 0]$
          Request accepted and routed on $P$

Case 2: $\boldsymbol{e =}$ Departure of $\boldsymbol{b}$ from LSP
      If $B_P > B_{Th}$
        Re-dimension LSP.                 $s_{m+1} = [0, B_L + B_P - b, 0]$
      Else
        If $B_P = 0$ and $B_L = b$
          Tear-down LSP.                  $s_{m+1} = [0, 0, 0]$
        Else
          Do not re-dimension LSP.        $s_{m+1} = [A + b, B_L - b, B_P]$
```

Figure 8: Least One-Step Cost Algorithm.

optimal policy table needs to be updated when there are changes in the network topology. The update can, however, be performed off-line. For networks of considerable size, the storage of the optimal policy for each node pair can be very resource-consuming. In such cases the sub-optimal policy, given in Section 3.5, can be applied. This policy computes the decision upon arrival of each request and does not involve storage of the whole policy.

In this section, the performances of both the optimal policy (decision rule in Eq. 21) and the sub-optimal policy (decision rule in Eq. 25) are presented and compared. The performance metric is the *discounted total cost* defined in Eq. 8. Both the optimal and the sub-optimal policies can also be compared with heuristics where LSP optimization is absent or performed either for each event arrival or periodically.

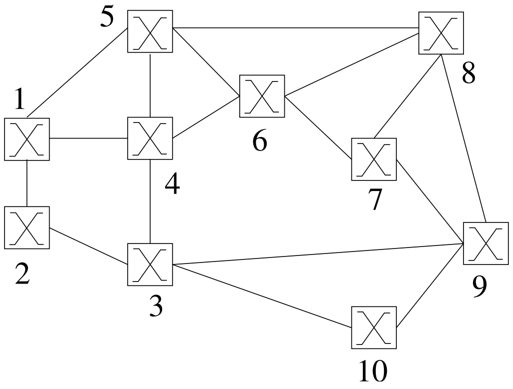


Figure 9: Network Physical Topology G_{ph}.

Table 3: Cost Coefficients.

c_{s}	c_{a}	c_{b}	c_{ip}	c_{mpls}
15	15	1	2.5	0.5

3.6.1 Simulation Model

For the simulations, the MPLS network is modeled as a non-hierarchical graph G_{ph} shown in Figure 9. It is a 10-node random graph with a maximum node degree of 3 and 17 edges. Each node represents an LSR and each edge represents a physical link connecting two LSRs. Each link is assumed to have a physical capacity of 1000 c.u.. Based on this network model, the adjacency matrix of the network as well as the number of links of the shortest path between any two nodes can be obtained apriori. The number of links in the shortest path between any node pair estimated by the source is deterministic. The request duration is assumed to be exponential whereas the request arrival arrival follows a Poisson process. The values given in Table 3 are assumed for the cost coefficients in Eqs. (5)-(7) which define the cost incurred by the network. With these cost coefficients, the threshold $B_{Th}(i,j)$, defined in Eq. 24, for the sub-optimal policy (decision rule in Eq. 25) becomes

$$\begin{aligned} B_{Th}(i,j) &= \frac{15(h+1)(\alpha+\lambda+\mu)}{2(h-1)} \\ &= \frac{7.5(\alpha+\lambda+\mu)(h+1)}{(h-1)}. \end{aligned}$$

For different cases, the values of λ and μ are varied to obtain the $B_{Th}(i, j)$ independently. In all the simulations, the user bandwidth requests are assumed for the amount of 1 c.u.. Even though both the optimal and sub-optimal policies are independent of the amount of the bandwidth requested, this homogeneous case is useful because the results obtained are representative of the effects the bandwidth requests have on the MPLS network topology. When the bandwidth requests are for 1 c.u., a snapshot of the events can be obtained to really understand how the events are triggered.

For each source and destination pair, the value iteration algorithm of Figure 7 is used to determine the minimum discounted total cost (defined in Eq. 8) and the optimal policy. For the value iteration algorithm, ε is set to 0.1% of the first-step discounted total cost. The minimum discounted total cost is then averaged over all possible source and destination pairs. For the proposed sub-optimal policy also, the minimum discounted total cost is calculated using the value iteration algorithm. As given in Eqs. (5)-(7), the cost functions are linear with respect to the bandwidth requests.

3.6.2 Results

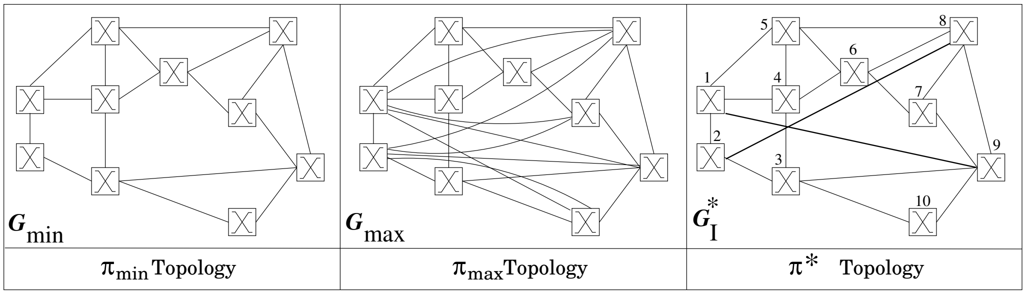
The following simulations demonstrate the performance of the two policies. It is shown how high traffic volume leads to LSP setup/re-dimensioning whereas for less volume, the LSPs are not modified. The MPLS network topology is modified according to varying bandwidth requests. Various experiments demonstrate cases where the results of the two policies are different and then compare their performance.

Six different traffic loads with different characteristics are offered to the network to analyze the performance. The λ/μ values for the six cases are given in Table 4. These node pairs were selected because they have two or more hops between them.

For experiment I, the requests arrive with λ_1/μ_1 and the optimal policy π^* is applied. The resulting MPLS network G_I^* is shown in Figure 10(c). Note that since the node pairs 1-9 and 2-8 have a traffic load greater than the others, representing a focused overload

Table 4: Bandwidth Requests for Experiments

Node pair	1-7	1-8	1-9	1-10	2-7	2-8	2-9	2-10	Others
λ_1/μ_1	5	5	30	5	5	30	5	5	0
λ_2/μ_2	5	5	5	30	5	5	5	5	0
λ_3/μ_3	5	5	5	5	5	5	5	5	0
λ_4/μ_4	10	10	10	10	10	10	10	10	0
λ_5/μ_5	30	30	30	30	30	30	30	30	0
λ_6/μ_6	50	50	50	50	50	50	50	50	0



(a) (b) (c)

Figure 10: Topologies and Costs for (a) $\pi_{\min}$, (b) $\pi_{\max}$ and (c) π^*.

scenario, the corresponding LSPs have been established. Instead, if the proposed sub-optimal policy $\pi^{\#}$ (decision rule in Eq. 25) is applied, the resulting network $G_I^{\#}$ coincides with G_I^*, demonstrating the efficiency of the sub-optimal policy. In Figure 10(a) and (b) are shown, for comparison, $G_{\min}$ and $G_{\max}$ that would result if the two simple heuristic decision policies $\pi_{\min}$ and $\pi_{\max}$ were applied, respectively. $\pi_{\min}$ is the policy to never establish non-default LSPs whereas $\pi_{\max}$ is the policy to adapt the LSP to each occurring event. It is found that the discounted total cost (defined in Eq. 8) for G_I^* is 45% lower than $G_{\min}$ and 77% lower than $G_{\max}$.

Experiment II aims to verify the on-line adaptability of the optimal policy π^* (decision rule in Eq. 21) when a traffic variation occurs. Now, the requests arrive with λ_2/μ_2 given in Table 4. If the optimal policy is applied starting from the initial state represented by G_I^*, the result of experiment I, the final topology consists of an added LSP(1,10) to G_I^*.

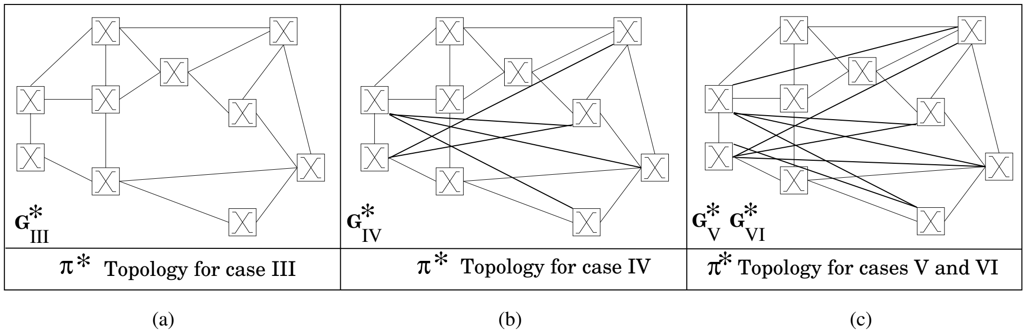
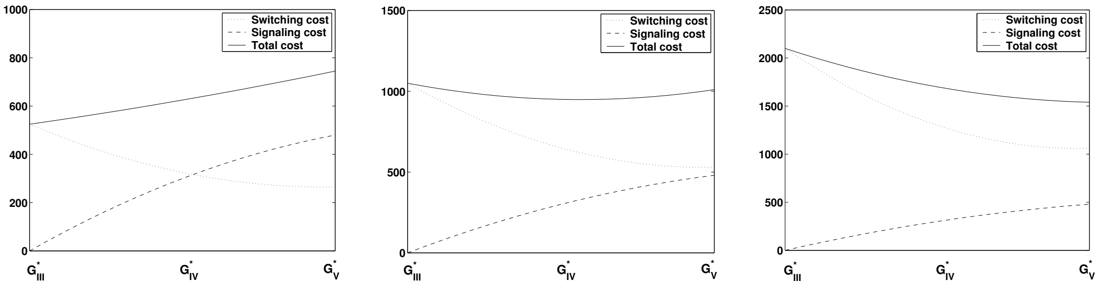


Figure 11: Topologies for Experiments III-VI.

The old non-default LSPs are not torn-down because they are utilized by the traffic as they provide reduced switching cost (Eq. 7) without the overhead of the signaling cost (Eq. 5). The topology has changed from G_I^* to better fit the new traffic pattern. On the other hand, if the initial topology was G_{ph} then the final network topology will just add the LSP(1,10) to G_{ph}. So, the resulting topologies in the two cases differ and highlight the capability of the optimal policy to adjust to the traffic variation. Same results are obtained upon application of the sub-optimal policy $\pi^{\#}$ (decision rule in Eq. 25).

Starting with the initial topology G_{ph}, the traffic matrix was homogeneously increased for experiments III-VI as shown in Table 4. The corresponding π^* topologies are shown in Figure 11. As expected, for larger bandwidth requests, more LSPs are set-up because the expected bandwidth and switching costs (Eqs. 6 and 7) exceed the signaling cost (Eq. 5) overhead and it becomes economically viable to setup the LSPs. More LSP setup leads to a more connected MPLS network: the network G_{V}^* is more connected than the network G_{IV}^*, which is in turn more connected than the network G_{III}^*. However, once the traffic exceeds the threshold, the LSPs are setup. Thus, the final topologies G_{V}^* and $G_{\text{V}I}^*$ for the experiments V and VI are similar. If the sub-optimal policy (decision rule in Eq. 25) is applied, slightly different results are obtained. For experiment III, $G_{\text{III}}^{\#}$ is same as G_{III}^* because the traffic is very less and it is not economically efficient to set-up any LSPs. For experiment IV, the sub-optimal policy does not find it viable to set-up any LSPs and hence



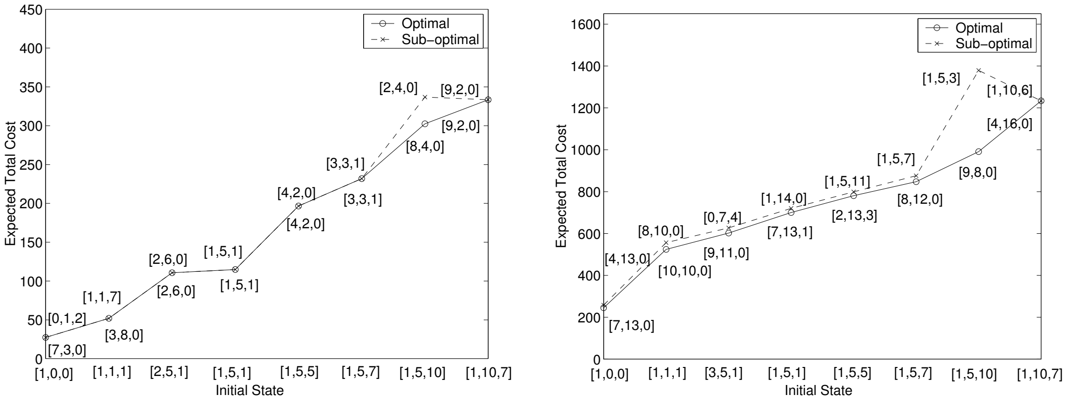
(a) Experiment III (b) Experiment IV (c) Experiment V

Figure 12: Total Cost and Cost Components for Experiments (a) III, (b) IV and (c) V.

$G^{\#}_{\text{IV}}$ does not add any new LSPs, *i.e.* it is same as G^*_{III}. For experiment V, the traffic is a little higher and thus, the threshold B_{Th} in Eq. 24 is exceeded for LSPs with length 3 but not for LSPs with length 2. Thus $G^{\#}_{\text{V}}$ is same as G^*_{IV}. Finally, $G^{\#}_{\text{VI}}$ is the same as $G^*_{\text{V}I}$ as the threshold B_{Th} is exceeded even for LSPs with length 2. It can be seen from Eq. 24, the threshold is smaller for longer LSPs as B_{Th} is inversely proportional to $(h-1)$.

As a verification for the results in Figure 11, the costs of the topologies G^*_{III}, G^*_{IV} and G^*_{V} are calculated. The switching and signaling costs and the total discounted cost for the three topologies is shown in Figure 12. In each figure, the respective minimum discounted total cost corresponds to the topologies shown in Figure 11. For instance, in Figure 12(a), the minimum discounted total cost is given for topology G^*_{III}; in Figure 12(b), the minimum discounted total cost corresponds to topology G^*_{IV} and in Figure 12(c), the minimum discounted total cost corresponds to topology G^*_{V}.

Having seen a case where the final topologies obtained by application to policies π^* and $\pi^{\#}$ are different, their performance is not compared. For the initial topology G_{ph}, the total discounted cost for different initial states for one node-pair with three hops in between is shown in Figure 13. The final state of the system is shown for each initial state and the two policies as the numbers in brackets close to the lines. As the discount rate α (from Eq. 8) is smaller for Figure 13(b), the costs are larger in magnitude. It is seen that the expected costs are identical or marginally close, except for one point in each figure. For the initial



(a) α=0.5, λ/μ=5

(b) α=0.1, λ/μ=10

Figure 13: Total Expected Cost vs. Initial State.

state [1, 5, 10], the optimal policy optimizes the LSP immediately whereas the sub-optimal policy does not since the threshold is not exceeded, resulting in the lower expected cost for the optimal policy. On the other hand, for the initial state [1, 1, 1] in Figure 13(a), only the optimal policy performed the optimization but the costs are equal for both cases. This is because of the discount factor α as events too far in the future have marginal effect on the cost. One point to be observed from the figures is that the final states from the optimal policy have large available bandwidth values. This is because the optimal policy performs LSP optimization very often whereas the sub-optimal policy performs optimization only when the traffic exceeds a threshold which is large. This, in effect, reduces the sensitivity of the decision policy to minor variations in the traffic, *i.e.*, by filtering small fluctuations.

In Figure 14, a stepwise increased homogeneous traffic is offered to observe the percentage setup of LSPs using the sub-optimal policy $\pi^{\#}$. For λ/μ values less than 10, no LSP is setup as no threshold is exceeded. A stable configuration of the network is achieved for λ/μ values between [20,30] where all LSPs with length 3 are setup and those with length 2 are not setup. For λ/μ greater than 45, all the LSPs are always setup and the network reaches its fully connected stable state. For the other values of λ/μ, the LSPs are setup with percentages as shown in Figure 14, *e.g.* for λ/μ of 15, the LSPs with length 3

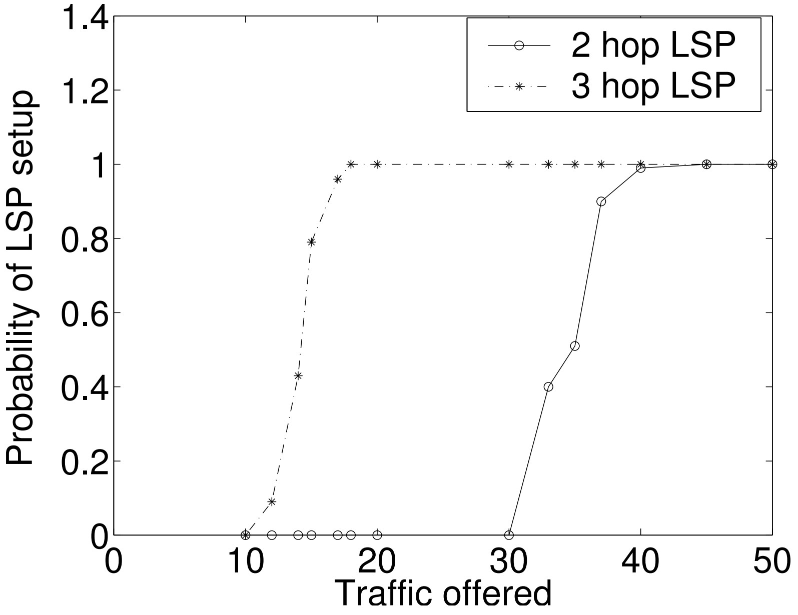


Figure 14: Percentage Setup of LSPs for Homogeneous Traffic.

are setup with 80% probability.

The simulations show that the value iteration algorithm is very efficient and stable. The convergence is fast resulting in a low number of iterations. In general, the number of iterations does not depend on the cost parameters $(c_{\mathrm{s}}, c_{\mathrm{a}}, c_{\mathrm{b}}, c_{\mathrm{ip}}, c_{\mathrm{mpls}})$, but depends on the values of λ and μ. There are other iteration algorithms, (*e.g.* , policy iteration [91]) that have a higher rate of convergence but are more intensive computation-wise (the policy iteration involves a search through the set of all possible decision policies). The proposed sub-optimal policy is much less computationally intensive (no storage of decision policy) and provides the expected discounted total cost values close to the optimal policy.

Chapter 4

QoS Estimation Based Path Selection

Efficient routing of user requests is essential in a DiffServ/MPLS network to satisfy QoS and manage network resources. In this chapter, a novel algorithm is presented that finds a feasible path that minimizes the cost incurred. The cost is attributed to bandwidth carriage, and switching and signaling efforts in the network for the requested connection. This algorithm was introduced in [92].

This chapter is organized as follows: The motivation for the development of the path selection algorithm is given in Section 4.1. In Section 4.2, other traffic routing algorithms are described. Then, in Section 4.3, the formulation for the path selection algorithm is presented. The algorithm to predict the metrics in case of partial information is given in Section 4.4. Performance evaluation of the path selection algorithm is provided in Section 4.5.

4.1 Motivation

To support Quality of Service (QoS) in an MPLS network, many algorithms have been designed for various network components, such as scheduling, admission control and routing. However an efficient and scalable algorithm for QoS routing of traffic flows is still missing, even if much research has been focused on this subject. The goal of QoS routing is to find paths that have sufficient resources to satisfy the QoS requirements of a request. A scalable QoS routing algorithm has to work efficiently for large scale networks and, for this purpose, many practical and theoretical issues have to be solved.

Usually a request requires more than one metric to be considered for routing purposes. For example, real time applications like Internet telephone, distributed games and video conference have multiple QoS requirements on delay, cost, delay jitter, loss ratio, bandwidth, etc. In this sense, multi-constrained routing deals with finding a path that satisfies multiple QoS constraints on diverse metrics. It is well known that this problem is NP-complete [93], so finding an accurate and simple heuristic is one important characteristic of an efficient QoS routing algorithm. Another important issue to be addressed in the design of a QoS routing algorithm is the presence of inaccurate global network state information. In large scale networks, maintaining precise global state information is not possible due to the large amount of information needed and the delay incurred to flood it to the whole network. Thus, an effective QoS routing algorithm must be able to work properly even while using inaccurate network state information. Last important issue is the relation between the QoS routing and resource reservation protocol. In fact, if the reservation is performed flow by flow (using for example RSVP), QoS can be provided with a high degree of accuracy but maintaining per flow information in the routers leads to scalability problems. On the other hand, using a stateless QoS framework like DiffServ is scalable but can not provide, by itself, QoS guarantees. The addition of the MPLS architecture [25] or a tunneling and encapsulation ad-hoc technique [94] to the DiffServ classic architecture provides the right framework for scalable QoS routing, allowing aggregated resource management.

4.2 Related Work

QoS routing is an extensively studied subject [36]. It has come a long way from the simple Dijkstra routing [37]. Much of the work in the field of QoS routing has concentrated on the delay constrained least cost problem [38, 39, 40]. Since the problem is NP-complete, the proposed solutions are heuristic in nature [41]. Some effort has also concentrated towards heuristic algorithms based on Lagrangian relaxation [42, 43]. The relaxation approach is based on an aggregate weight (obtained by summing the additive metrics) which is used in the Dijkstra algorithm for route computation. This approach does not have the capability to consider non-additive metrics. In MPLS networks, the routing research has concentrated on LSP routing *i*.e. how to route the LSPs in the network. Many schemes such as Minimum Interference Routing Algorithm (MIRA) [44], Profile Based Routing (PBR) [45] along with modifications to OSPF, IS-IS have been proposed for LSP routing. However, a scheme for routing of traffic flows in an MPLS network is not considered.

With few exceptions, previous QoS routing solutions have been developed under the assumption that the exact state of the network is known to nodes performing the route computation. In practice, however, network state is not known for certain due to the following reasons [95, 96]. First, current link-state routing protocols such as OSPF flood link values periodically. To limit the overhead of flooding, long update intervals are used. A second source of inaccuracy is attributed to state aggregation. Most link-state routing protocols are hierarchical, whereby the state of a group of nodes (an OSPF area or a PNNI peer group) is summarized (aggregated) before being disseminated to other nodes.

4.3 Problem Formulation

In this chapter, a QoS routing algorithm for traffic flows in MPLS networks is presented. This routing algorithm is unique because of the dynamic nature of the MPLS network topology. The algorithm considers multiple metrics, is scalable and operates in the presence of inaccurate information. Numerous path choices are compared in terms of their operational costs. The cost considers all the metrics important for the path selection. The factors pertaining to the different metrics are weighed by their corresponding importance factor which can be varied from network to network. In essence, the novelty of the proposed algorithm lies in the cost structure for the LSPs and the ability to deal with partial network information.

QoS routing is typically attributed to cause additional costs to the network due to the added computational cost and protocol overhead. The computational overhead can be compensated by upgrading the network components' processing power. To reduce the protocol overhead, many solutions have been suggested to reduce the frequency of link state update.

This reduced frequency leads to partial network information at each node.

4.3.1 Partial Information

The protocol overhead is introduced due to the increased amount of network information needed at all the network nodes to make QoS-aware routing decisions. The overhead is highest if information about each network state update is flooded throughout the network. There are essentially three ways to control the volume of link state update traffic in a network. The first method controls the criterion for triggering an update, reducing the frequency of updates. According to this control method, the network information is not flooded for each, possibly insignificant, network state update, rather some triggering policy is used to generate updates at a reduced frequency. The second method restricts the flooding of the update in the network, reducing the propagation of updates. In this way, the updates are not flooded throughout the network but only to the nearest neighbors. The information is propagated through the network hop-by-hop over time. The third method updates only aggregated topology information in the network, reducing the content of updates. The updates convey information that is aggregated for a path or a sub-area in the network in the form of time average etc.

All the three methods are widely-studied and accepted for generating partial network state information for routing algorithms [97, 98]. The second method of restricted flooding faces the problem of increased convergence times as the information propagates only hop-by-hop at each update instant. The third method of aggregated information is highly scalable but it is not efficient in providing QoS satisfaction to requests as only aggregated information is known about the network. Thus, the first method of controlling the update frequency seems to be the best approach. It is scalable and more efficient. A variety of triggering policies can be used for this purpose. Obviously, the sensitivity of the triggering policy is directly related to the accuracy of the information available at the routers. The more aggressive the triggering gets, more is the information available at the routers. However, an aggressive triggering policy would lead to increased update traffic and would beat the purpose of triggered updates completely. Thus, the triggering policy has to be chosen judiciously. There are three main update triggering policies: periodic, threshold based, and class based policies. Periodic policy maintains a timer and when the timer elapses, all the network nodes update the network with their state. For the threshold based update, the update is triggered if the difference between the current metric value and the last advertised value is greater than a threshold. The class based update policy divides the complete range of values, that the metric can take, into classes. The class size can be equal or exponential. If the class of the current metric value is different from the class of the previously advertised value, then the update is triggered.

Each of these update policies has its own advantages and disadvantages. The periodic policy is simple to implement but it may not give a clear idea about the current network state. In other words, the metric may change its value drastically but the update may not be triggered because the timer is not yet expired. The threshold and class based policies are more difficult to implement in a real network since the previous state needs to be remembered. In the case that the metric value oscillates close to the class boundary, the class based method will generate updates for each variation and thus will be flooding the network with the updates. However, in general, both the threshold and class based policies result in better information at each node about the updates of the network. The periodic update policy is chosen in this thesis to provide partial link state information in the network. The periodic policy is chosen because of its simplicity. An estimation and forecast algorithm will be used to compensate for the lack of accurate information.

4.3.2 Network Model and Costs

In this formulation, the problem of QoS routing in IP networks is handled. The goal of QoS routing is to find a low-cost feasible path that has enough available bandwidth, while restricting the number of hops and the delay on the path. The approach of [94] will be used, using tunneling and packet encapsulation with the relay nodes. However, the metric is chosen for each tunnel based on cost minimization.

The network is represented with a directed graph $G(N, \mathcal{L})$. Here, N is the set of nodes in the network and $\mathcal{L}$ is the set of LSPs. Each LSP LSP$(i, j) \in \mathcal{L}$ corresponds to an LSP between the nodes i and j ($i, j \in N$) and is assigned a cost C^l_{ij}. A *path* p_{uv} between the nodes u and v ($u, v \in N$) is defined as a concatenation of LSPs LSP$(u, x), \ldots,$ LSP(z, v), where the nodes $x, \ldots, z$ are arbitrary nodes in the network. These nodes are called the *relay nodes* on the path p_{uv}. When the path p_{uv} coincides with just one LSP, p_{uv} is called as a direct path between nodes u and v and the functionalities of the relay node are not needed. There will be many paths p_{uv} between any node pair u and v, including the direct path. Let $\mathcal{P}_{uv}$ denote the set of all such paths. A cost C^p_{uv} is associated with the path p_{uv} and LSP$(i, j) \in p_{uv}$ denotes that LSP(i, j) belongs to the path p_{uv}. The following quantities can be defined:

- A^l_{ij} : Available capacity on LSP(i, j)
- d^l_{ij} : Delay incurred on LSP LSP(i, j)
- A^p_{uv} : Available capacity on path p_{uv}
- d^p_{uv} : Delay incurred on path p_{uv}
- n^p_{uv} : Number of LSPs in path p_{uv}

A^p_{uv} is the minimum of the available capacities A^l_{ij} of all the LSPs comprising the path p_{uv}. The path delay is assigned to be equal to the sum of the delays on the individual LSPs in the path. In other words,

$$A^p_{uv} = \min_{\text{LSP}(i,j) \in p_{uv}} A^l_{ij}$$

$$d^p_{uv} = \sum_{\text{LSP}(i,j) \in p_{uv}} d^l_{ij}$$

A framework is needed to compare and choose among all the feasible paths between a node pair. Towards this end, costs are associated with the LSPs and paths. The cost is attributed to five factors: bandwidth requested, switching, signaling, remaining available bandwidth and delay.

The rate at which the bandwidth cost is incurred on LSP(i, j) depends linearly on the bandwidth required by the connection. Thus, W_{ij}^{b}, the bandwidth component of the cost can be written as

$$W_{ij}^{\text{b}} = b\, c_{\text{b}}^{l}\, T \tag{26}$$

where c_{b}^{l} is the bandwidth cost coefficient per capacity unit (c.u.) in the network, b is the bandwidth requested by the connection and T is the time duration for which the connection is valid. The rate of the switching cost on the LSP is also proportional to the requested bandwidth. Thus, the switching cost component can be written as

$$W_{ij}^{\text{sw}} = b\, c_{\text{sw}}^{l}\, h\, T \tag{27}$$

where c_{sw}^{l} is the switching cost coefficient per capacity unit (c.u.) in the network and h is the length of the LSP. This component is also proportional to the duration of the connection because the switching cost has to be paid for each packet of the connection as long as it holds. On the other hand, the signaling cost is a one-time cost to signal the setup of the path over the LSP. Thus, the signaling cost is given as

$$W_{ij}^{\text{sign}} = c_{\text{sign}}^{l} \tag{28}$$

where c_{sign}^{l} is the signaling cost coefficient in the network. This value is independent of the amount of bandwidth requested as it corresponds to the signaling effort, which is performed for connection establishment. Also, it is independent of the connection duration because this cost is incurred during the connection establishment. The next factor contributing to the cost is the available bandwidth left on the LSP after the connection has been granted. This cost is given as

$$W_{ij}^{\text{AB}} = \frac{c_{\text{AB}}}{A_{ij}^{l} - b} T. \tag{29}$$

Such an inverse structure is chosen for the available bandwidth cost since the available bandwidth is not a linearly additive metric (like hop, delay) in the network and LSPs with less available bandwidth are assigned a higher cost. The last term in the LSP cost comes from the delay incurred on the LSP. This cost is given as

$$W_{ij}^{\mathrm{d}} = c_{\mathrm{d}}\, d_{ij}^{l}\, T. \tag{30}$$

Summing up all these individual costs, the total cost incurred for successfully granting the requested bandwidth on LSP(i, j) can be calculated. However, weighting factors should be used for these costs to modify the importance given to the components. A higher weighting factor would imply a higher relative significance of the associated cost component. Thus, the total cost is given as:

$$W_{ij}^{l} = \alpha \,\{\, W_{ij}^{\mathrm{b}} + W_{ij}^{\mathrm{sw}} + W_{ij}^{\mathrm{sign}} \,\} + \beta\, W_{ij}^{\mathrm{AB}} + \gamma\, W_{ij}^{\mathrm{d}} \tag{31}$$

Notice that the first three cost components have a single weighting factor. This is because all three of them relate to the distance metric and should be weighed identically.

A path in the network is a concatenation of LSPs. If the path includes just one LSP, its cost is equal to the LSP cost. However, if the path is composed of two or more LSPs, its cost is not just the sum of the individual LSP costs. This is because the relay nodes between the LSPs have to perform additional switching and signaling due to the change in the encapsulation from one LSP to the other. Thus, the path cost is given as:

$$W_{uv}^{p} = \sum_{l_{ij} \in p_{uv}} W_{ij}^{l} + R(c_{\mathrm{sw}}^{p}\, T\, b + c_{\mathrm{sign}}^{p}). \tag{32}$$

Here, c_{sw}^{p} and c_{sign}^{p} denote respectively the coefficients for the IP switching and signaling costs incurred due to the presence of relay nodes in the path and there are R relay nodes.

The cost coefficients are introduced in the cost definitions above to provide a relative weight to each of the cost components. A network operator can decide these coefficients based on the fraction of the total cost that is attributed to each cost component. As mentioned before, the objective of QoS routing is to find a feasible path with enough available bandwidth while satisfying the delay and hop constraints. The proposed algorithm tries to achieve a balance between maximizing available bandwidth and minimizing the number of hops and delay.

The exact path selection problem can be specified as:

$$p^\star_{uv} : W^{p^\star}_{uv} = \min_{p \in \mathcal{P}_{uv}} W^p_{uv} \tag{33}$$

subject to the feasibility constraints

$$n^{p^\star}_{uv} \leq k,$$

$$d^{p^\star}_{uv} \leq d_{\text{max}}$$

and

$$A^{p^\star}_{uv} \geq A_{\text{min}}.$$

A concession of k units is allowed in the length of the chosen path w.r.t. the direct LSP to be able to consider paths which are a few hops longer than the shortest path. d_{max} denotes the maximum allowed delay and A_{min} denotes the minimum required available bandwidth on the path by the flow. A_{min} is assumed to be larger than the bandwidth requested by the flow. This is based on two reasons. Firstly, LSPs should not be fully occupied and secondly, the current actual values of A^l_{ij} may not be the most recently advertised value. Thus, the cushion in A_{min} over the bandwidth requested by the flow is used to compensate for the information uncertainty.

The proposed algorithm provides a heuristic to the exact path selection procedure by limiting the number of paths to be considered to F instead of an exhaustive search. A combination of various metrics is used for the path selection. The cost structure defined earlier provides a framework to choose the most efficient path for the traffic flow.

The operation of the algorithm is as follows. The centralized manager TEAM tries to find F paths between the source and the destination. These paths are obtained by increasing the hop count in the path. In other words, if there is a direct LSP between the source and destination, it is a candidate for consideration. Next, all paths with 2 LSPs between the source and destination are considered. If such paths exceed F-1 in number, then the first F-1 paths are chosen to be candidates. Note that these paths have been found without any consideration for feasibility. If still F candidates are not found, then the search proceeds to include paths which have 3 LSPs. The search goes on in this manner by increasing the number of LSPs in the path, till F candidate paths are found. These F paths are then checked for feasibility against the constraints specified in Eq. 33. The feasible paths are then the final set of candidates for routing the traffic flow. The total cost (defined in Eq. 32) of these paths is then evaluated and compared. The least cost feasible path is then chosen for the traffic flow.

This algorithm assumes knowledge of the exact values of the metrics associated with all the LSPs in the network. However, in reality, the metric updates are not instantaneous for large networks. The finite update time can compromise the scalability of the proposed routing algorithm. However, the algorithm can be made scalable by modifications that can operate in the presence of inaccurate/partial information about the network state and still providing comparable performance.

4.4 Partial Information

Each network node floods information about its state to the whole network at periodic time intervals. This results in partial information about the network state at each node. The information is partial because the network nodes do not have current information about the complete network state. Instead, they have information for the time instant when the last update was generated. When a path selection request arrives, an estimation and forecast algorithm can be used to obtain more accurate information about the current network state. This algorithm can be applied to estimate and forecast the available bandwidth as well as delay of an LSP.

Let p be the number of past samples that are used in the prediction and L the metric that is being predicted. The arrival of the path selection request is not synchronous with the update period. This means that the time interval between the instant at which the estimation is required and the arrival of last update is less than the update periodicity. So, at the instant of estimation and forecast, the past p samples $\{L_1,\, L_2\, \ldots,\, L_p\}$ are known and the next sample L_{p+1} can be forecasted using a linear prediction:

$$L_{p+1} = \sum_{n=1}^{p} L_n\, w_n \tag{34}$$

where on the right side are the past samples and the prediction coefficients w_n and on the left side, the predicted value. The formulation can be rewritten as $L_{p+1} = \mathbf{L}\, \mathbf{w}^T$, where $\mathbf{L} = [L_1,\, L_2,\, \ldots\, , L_p]$ and $\mathbf{w} = [w_1,\, w_2,\, \ldots\, , w_p]$. The problem can be solved in an optimal manner using covariance method. The parameter p can be dynamically changed based on the forecast performance. The covariance can be estimated from the available measurements as $r_L(n, m) = \sum_{i=p-M}^{p} L_{i-n} L_{i-m}$ where M affects the accuracy of the estimation, *i.e.*, more the samples considered, more precise the estimation is. The number of samples needed for a given n and M is $(n + N)$. The solution of the covariance equation will provide the prediction coefficients for the Eq. 34. If the time instant k, where information is desired, is closer to p than to $p + 1$, then L_p is the metric estimation at instant k. However, if k is closer to $p+1$, then the estimated value as $\left[\hat{L}_{p+1}\right]$. To further improve the prediction performance, an update algorithm to adjust the value of p can be used. According to this algorithm, if the ratio of the prediction error e to the actual value L_{p+1} is above a threshold e_{Th}, which implies that the error is large w.r.t. the actual value and the prediction performance was not too good, increase the value of p to consider more samples in the prediction process. If the ratio is smaller than e_{Th}, then reduce the value of p. However, an upper bound p_{max} is put on p because large values of p increase the computational cost of the regression. The threshold is determined based on the traffic characteristics and the conservatism requirements of the network domain. It represents the confidence in the estimation procedure in terms of prediction errors. By using this prediction approach, the drawback of the periodic update approach related to the unresponsiveness to significant metric changes is eliminated.

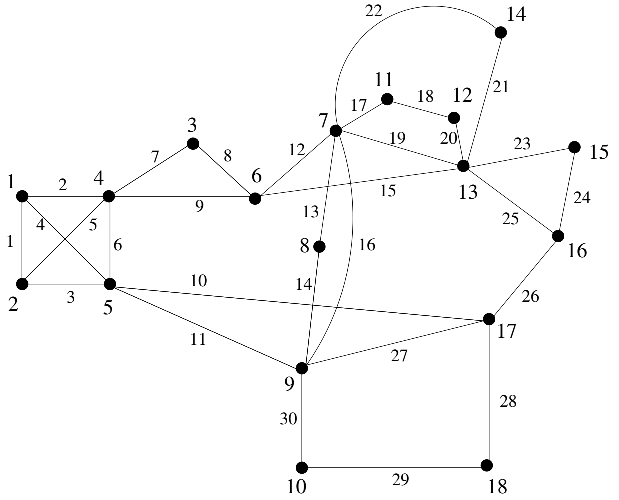


Figure 15: Network Topology.

4.5 Performance Evaluation

In this section, the proposed path selection algorithm is compared with other well-known algorithms, from the viewpoint of performance and robustness. Extensive simulations were conducted to evaluate the performance and computational complexity of the proposed algorithmic solution. The goal of these simulations is to evaluate the goodness of the proposed algorithm and to demonstrate the benefits of the approach of using multiple metrics for path determination without a loss of routing performance.

The topology of Figure 15 was used for the simulation experiments. This represents a popular "isp" topology used in many QoS routing studies [97] and is typical of the nation-wide network of a US based ISP. All the links are bidirectional with a capacity of 155 c.u. in both directions. k, the number of extra hops allowed in the feasibility constraint of the path selection problem in Eq. 33, was set to 5 in order to allow for longer paths w.r.t. the shortest path. A_{min} in the path selection problem is constrained to be at least 10 c.u. above the bandwidth requested and the number of paths considered for cost comparison in each algorithm was restricted to F, which is set to 20. This value was chosen so that enough number of paths distinct from the min-hop path are selected, and at the same time an exhaustive search is not necessary for the paths. Unity values were assigned to each of the cost coefficients and also to the three weighting factors α, β and γ in the cost formulation of Section 4.3.2. Cost coefficients are special quantities that vary from network to network depending on the parameters important to the network. The weighting factors are used to assign different weights to different components. By suitable choice of the weighting factors, the routing performance of several well-known routing algorithms can be obtained. The weighting factors can be adapted to the traffic load in the network. For example, if the network is lightly loaded, the shortest path routing can give a satisfactory performance. The shortest path algorithm can be obtained by setting $\alpha = 1$ and $\beta = \gamma = 0$. Since the network parameters and conditions are unknown, the cost coefficients were set to unity. In this way, the simulations are impartial to cost components. The weighting factors were also assigned unity values to give equal importance to each cost metric.

25 independent experiments were performed with the network of Figure 15. Traffic was introduced from nodes on the left side of the network (1,2,4,5) towards the right side (11, 12, 13, 14, 15, 16, 17). In this way, focused overload was created in the middle of the network. In such a scenario, using shortest path routing algorithm can be penalizing as the network is overloaded. Thus, a more intelligent and efficient routing algorithm should be preferred which will give a better performance. This is confirmed by the results of Figure 16 where the rejection ratio of the three algorithms is compared to the shortest path routing algorithm. As can be seen, the three algorithms have reduced the rejection ratio by around 75% w.r.t. the shortest path routing.

Next, the algorithm's performance is compared w.r.t. the three metrics considered. The minimum available bandwidths for all the LSPs in the network is considered. These results are presented in Figure 17. As can be seen, the minimum available bandwidth is lower for the shortest path routing in contrast to the proposed algorithm. This is expected because the shortest path routing is limited to only one path between a node pair for every request, unlike the proposed algorithm that selects paths by considering a combination of metrics. On the other hand, the mean available bandwidth is larger for the shortest path routing. This

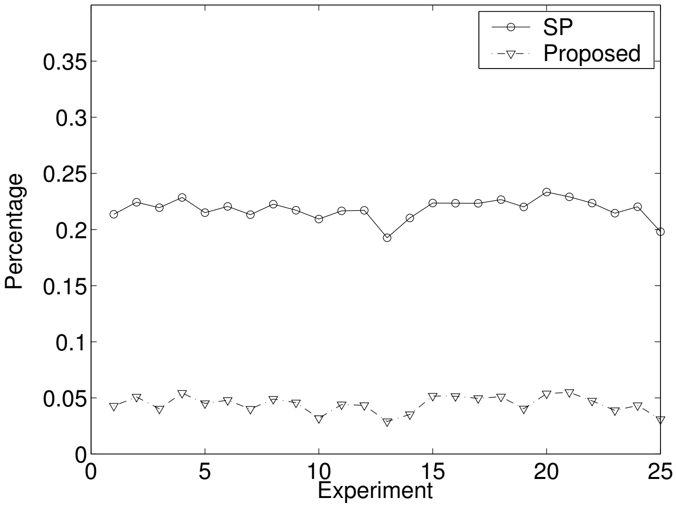


Figure 16: Rejection Ratio.

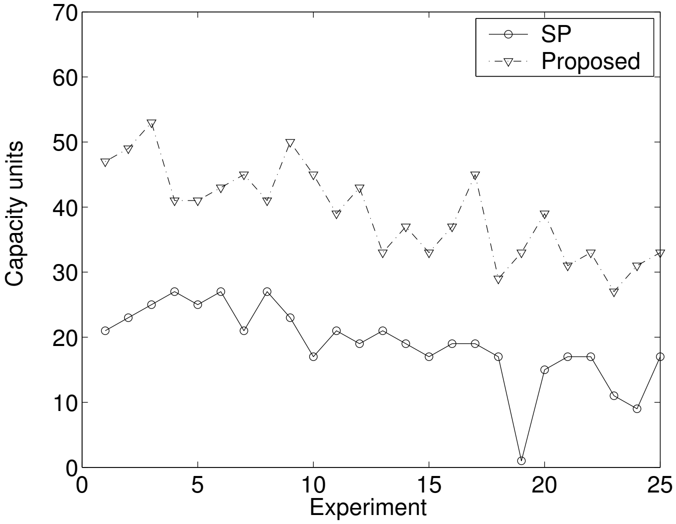


Figure 17: Minimum available bandwidth.

gives the false impression that the performance of the shortest path algorithm is better than the proposed algorithm. However, this is attributed to the poor load balancing achieved by the shortest path routing, as opposed to the proposed algorithm. The shortest path algorithm chooses the same path between a node pair every time it is executed. Thus, it has a high rejection ratio and the load is concentrated on a few LSPs. The proposed algorithm chooses possibly different paths for the requests (depending on the network state) and thus distributes the load in the network achieving a lower rejection rate.

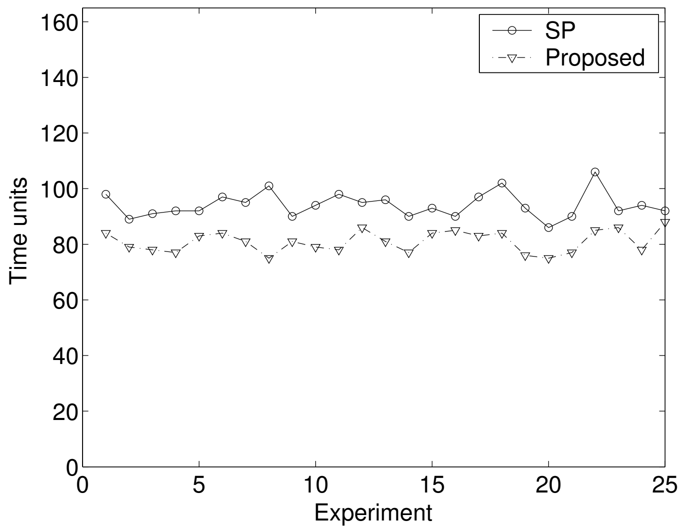


Figure 18: Minimum delay.

The second metric compared is the delay encountered by the request packets. The delay is composed of three components: the transmission, propagation and the queuing delay. The first two components are constant, however the queuing delay is determined by the load on the link. In other words, for a highly loaded LSP the queuing delay is larger than the value for a lightly loaded LSP. In Figure 18, the results of 25 simulation runs are presented for the delay encountered in the network. The minimum delay incurred by the shortest path routing is larger than the proposed algorithm. This is due to the over-loading of a few LSPs in the network.

The third metric is the number of paths with relay nodes. This number is obtained by taking a network snapshot at some time. The number of requests that were routed along paths with relay nodes was counted and it is shown in Figure 19. Obviously, the number of paths with relay is 0 for the shortest path routing whereas the proposed routing algorithm has a large number of paths with relay nodes.

With these results, the performance of the proposed algorithm has been compared with the shortest path routing algorithm. The shortest path algorithm was chosen as the basis for performance comparison because it is the current routing scheme in the Internet. By choosing appropriate values for the weighting factors α, β, γ in the proposed algorithm,

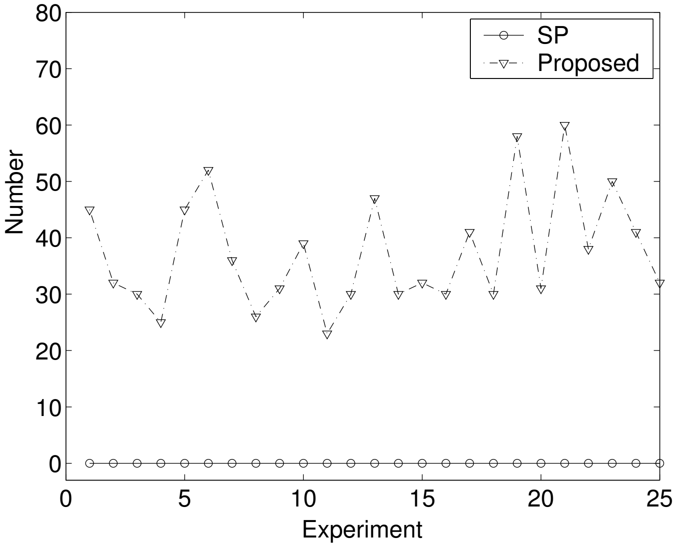


Figure 19: Number of paths with relay nodes.

other routing schemes and their results can be obtained. The performance of the proposed algorithm is superior to the shortest path routing.

In conclusion, the proposed routing algorithm is efficient and cost-effective. The cost is attributed to bandwidth carriage, and switching and signaling efforts in the network for the requested connection. The routing algorithm considers multiple metrics for path selection, and are scalable and operate under inaccurate network information.

Chapter 5

Available Bandwidth Estimation

With the growing traffic in the DiffServ/MPLS domain, tools are needed to understand the composition and dynamics of the traffic. In this chapter, an estimation algorithm for the available bandwidth on a link is presented. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. The algorithm dynamically changes the number of past samples that are used for prediction and also the the duration for which the prediction holds. This estimation algorithm was first introduced in [99] and later modified in [100].

This chapter is organized as follows: The motivation for the development of the available bandwidth estimation algorithm is given in Section 5.1. In Section 5.2, other measurement algorithms are presented. Then, in Section 5.3, the formulation for the periodic multistep prediction based estimation algorithm is given, followed by the non-periodic single-step prediction based algorithm in Section 5.4. The implementation considerations for the algorithms are presented in Section 5.5 and the performance evaluation in Section 5.6.

5.1 Motivation

Measurement is necessary for the network. A user would like to monitor the performance of his applications, check if level of service meets the agreement, etc. A service provider would like to monitor the current level of activity, enforce service level agreements (SLAs), plan for future etc. Few QoS metrics have been defined by the IPPM [101] working group of IETF. Some of these can be measured in the core of the network and others at the edges. Some have local significance at each router while others are end-to-end metrics. They can be obtained by measurements from various network elements. To obtain measured statistics from each network element is possible if individual users can monitor each such device. Due to security and privacy reasons, this is not possible in a network. Thus, common users can only measure the end-to-end metrics. The metrics with local significance at each router can only be measured by the network operators who can then make them publicly available. The approaches to monitor a network are *active* or *passive*. First gives a measure of the performance of the network whereas the latter of the workload on the network. Both have their merits and should be regarded as complementary. The active approach relies on the capability to inject packets into the network and then measure the services obtained from the network. It introduces extra traffic into the network. But the active approach has the advantage of measuring the desired quantities at the desired time. Passive measurements are carried out by observing normal network traffic, without the extra load. The passive approach measures the real traffic. But the amount of data accumulated can be substantial because the network will be polled often for information.

There are various quantities of interest that can be insightful about the state of the network. Available bandwidth (together with other metrics like latency, loss etc.) can predict the performance of the network. The *available bandwidth* of a link is the maximum throughput provided to a flow despite the current cross-traffic, when contrasted with the *capacity* which is the maximum throughput provided to a flow in absence of cross-traffic. Based on the bandwidth available, the network manager can obtain information about the congestion in the network, decide the admission control, perform routing etc. For MPLS networks, the available bandwidth information can be used to decide about the LSP setup [88], routing (Shortest Widest Path [93], Widest Shortest Path [102]), LSP preemption [103], etc. Each of these processes needs available bandwidth information at a suitable time-scale. It is desirable to obtain the available bandwidth information by measurements from the actual LSPs because they give more realistic information about the available bandwidth. The available bandwidth information could have been obtained by subtracting the nominal reservations from the link capacity which gives a lower bound.

In this chapter, an algorithm is proposed for estimating the available bandwidth of a link by dynamically changing the number of past samples for prediction and the number of future samples predicted with high confidence. The objective of the algorithm is to minimize the computational effort while providing a reliable estimate of available bandwidth of a link. It provides a balance of the processing load and accuracy. The algorithm is based on the dynamics of the traffic, *i.e.*, it adapts itself.

5.2 Related Work

The available bandwidth on a link is indicative of the amount of load that can be routed on the link. Obtaining an accurate measurement of the available bandwidth is crucial to effective deployment of QoS services in a network. Available bandwidth can be measured using both active and passive approaches. There are two definitions of available bandwidth. First one defines the available bandwidth on a single link (physical or virtual) of the network. This information can be used for congestion avoidance, routing etc. Second one defines the available bandwidth of a route on the network which manifests as the bandwidth measurement of the most congested link on the route. Various tools and products are available that can be used to measure available bandwidth of a link in the network. In [46], the authors have described a few bottleneck bandwidth algorithms. They can be split into two families: those based on pathchar [47] algorithm and those based on Packet Pair [48] algorithm. The pathchar algorithm is an active approach which leads to the associated disadvantages of consumption of significant amount of network bandwidth etc. The packet pair algorithm measures the bottleneck bandwidth of a route. It can have both active and passive implementations. Active implementations have bandwidth consumption whereas passive implementations may not give correct measurement. In [49], the authors have proposed another tool to measure bottleneck link bandwidth based on packet pair technique. Some other tools based on the same technique for measuring bottleneck bandwidth of a route have been proposed in [50, 51]. None of them measures the available bandwidth or utilization of a desired link of a network. In [52], the authors have proposed a tool to measure the available bandwidth of a route which is the minimum available bandwidth along all links of the path. It is an active approach based on transmission of self-loading periodic measurement streams. Another active approach to measure a path's available capacity is given in [53]. Iperf [54] from NLANR is another active approach that sends streams of TCP/UDP flows. Cisco has introduced the NetFlow [55] technology that provides IP flow information for a network. NetFlow provides detailed data collection with minimal impact on the performance on the routing device and no external probing device. But in a DiffServ environment, the core of a network is interested in aggregate rather than per-flow statistics, due to the scalability issues.

All the tools, except NetFlow, give path measurements based on an active approach. A network manager, on the other hand, is interested in finding the available bandwidth on a certain link of the network. It has access to the routers/switches of the network and can measure available bandwidth from the routers without injecting pseudo-traffic. Thus, it does not need the end-to-end tools that utilize the active approach of measurement. One approach is to use Simple Network Management Protocol (SNMP) [104] which is a short-term protocol to manage nodes in the network. An SNMP-managed network consists of three key components: managed devices, agents, and network-management systems (NMSs). A managed device is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information in Management Information Bases (MIBs) [105] and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers. An agent is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP. An NMS executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. Thus SNMP can be used as a passive technique to monitor a specific device. MRTG [106] is a tool based on SNMP to monitor the network links. It has a highly portable SNMP implementation and can run on most operating systems.

Thus, the network manager requires a tool for measuring the available bandwidth on a certain link of the network in a passive manner whenever it desires. Since the manager has access to the routers, it can use MRTG. But MRTG has the limitation that it gives only 5 minute averages of link utilization. For applications like routing, this large interval averaging may not be enough. MRTG can be enhanced to decrease the averaging interval down to 1 minute. This may still be large for some applications. Thus, in this thesis, MRTG has been modified to MRTG++ to obtain averages over 10 second durations. This gives the flexibility to obtain very fine measurements of link utilization. Even though the manager can have these measurements, it may not desire each measurement and also this will increase the load on the routers. So, a linear regression algorithm is proposed to predict the utilization of a link. The algorithm is adaptive because a varying number of past samples can be used in the regression depending on the traffic profile. The algorithm predicts the utilization and the reliability interval for the prediction.

5.3 Periodic Multi-step Prediction Based Algorithm

This algorithm predicts the link utilization by using linear regression for multi-step prediction with constant sampling frequency. With a constant sampling interval, a variable number of samples from the past are used in the linear regression to predict a variable number of samples in the future.

5.3.1 Problem Formulation

The most accurate approach for TEAM to measure available bandwidth will be to collect information from all possible sources at the highest possible frequency allowed by the

MIB update interval constraints. However, this approach can be very expensive in terms of signaling and data storage. Furthermore, it can be redundant to have so much information.

For a given periodicity for MRTG measurement, TEAM can measure the average link utilization statistic for that interval. The following notation for the link between two nodes i and j is proposed:

- C: Capacity of link in bits per sec,
- $A(t)$: Available capacity at time t in bits per sec ,
- $L(t)$: Traffic load at time t in bits per sec,
- τ: Length of the averaging interval of MRTG,
- $L_\tau[k]$, $k \in \mathbf{N}$: Average load in $[(k-1)\tau, k\tau]$.

The available capacity can be obtained as $A(t) = C - L(t)$. So, it would be sufficient to measure the load on a link to obtain available bandwidth. Note that the $i - j$ dependence of the the defined variables is not explicitly shown. This is because the analysis holds for any node pair independent of others. Also define:

- p is the number of past measurements in prediction,
- h is the number of future samples reliably predicted,
- $A_h[k]$: the estimate at $k\tau$ valid in $[(k+1)\tau, (k+h)\tau]$.

The problem can be formulated as linear prediction:

$$L_\tau[k+a] = \sum_{n=0}^{p-1} L_\tau[k-n]\, w_a[n] \qquad \text{for } a \in [1, h] \tag{35}$$

where on the right side are the past samples and the prediction coefficients $w_a[n]$ and on the left side, the predicted values. The problem can be solved using covariance method [107]. The values of p and h can be dynamically changed based on traffic dynamics. This distinguishes the proposed prediction method from other schemes based on linear regression.

```
ABEst Algorithm:
  1. At time instant $k$, available bandwidth measurement is desired.
  2. Find the vectors $w_a, a \in [1, h]$ using covariance method given $p$ and the previous measurements.
  3. Find $\left[\hat{L}_\tau[k+1], \ldots, \hat{L}_\tau[k+h]\right]^T$ using $[L_\tau[k-p+1], \ldots, L_\tau[k]]$ and Eq. 35.
  4. Predict $A_h[k]$ for $[(k+1)\tau, (k+h)\tau]$ .
  5. At time $(k+h)\tau$, get $[L_\tau[k+1], \ldots, L_\tau[k+h]]^T$.
  6. Find the error vector $[e_\tau[k+1], \ldots, e_\tau[k+h]]^T$ .
  7. Set $k = k + h$.
  8. Obtain new values for $p$ and $h$.
  9. Go to step 1.
```

Figure 20: The ABEst Algorithm.

5.3.2 Available Bandwidth Estimation Algorithm

The Available Bandwidth Estimation (ABEst) algorithm is given in Figure 20. p_0 and h_0 are the initial values for p and h. In step 2 of the algorithm, the covariance equations need to be solved. They are given in a matrix form as $\mathbf{R}_L w_a = r_a$, for $a = 1, \ldots, h$, where

$$
\begin{aligned}
\mathbf{R}_L &= \begin{bmatrix} r_L(0,0) & \cdots & r_L(0,p-1) \\ \vdots & \ddots & \vdots \\ r_L(p-1,0) & \cdots & r_{L(p-1,p-1)} \end{bmatrix} \\
w_a &= [w_a(0)\ w_a(1)\ \cdots\ w_a(p-1)] \\
r_a &= [r_L(0,-a)\ r_L(1,-a)\ \cdots\ r_L(p-1,-a)]
\end{aligned}
$$

In order to derive the covariance from the available measurements, it can be estimated as $r_L(n,m) = \sum_{i=k-N+p}^{k} L_\tau[i-n]L_\tau[i-m]$ where N affects the accuracy of the estimation, *i.e.* , the estimation is more precise if more samples are considered. The number of samples needed for a given n and N is $(n+N)$. Since the assumption about stationarity of the measurement sequence may not be accurate, the values of the covariance are updated every time there is a change in the value of p in step 8 of the algorithm. The solution of the covariance equations will provide w_a that can be used for predicting $\widehat{L}_\tau(k+a)$, $a = 1, \ldots, h$.

From the knowledge of the prediction coefficients w_a's, $\left[\widehat{L}_\tau[k+1], \ldots, \widehat{L}_\tau[k+h]\right]^T$ can be predicted using Eq. 35. Next step is to obtain an estimate of the available bandwidth for the interval $[(k+1)t,\ (k+h)t]$. This is done to obtain a single representative value valid for the whole interval. Two methods can be used for this, based on the requirements of the network manager. The representative available bandwidth value $A_h[k]$ can be given either as $A_h[k] = C - max\left\{\widehat{L}_\tau[k+1], \ldots, \widehat{L}_\tau[k+h]\right\}$ or by the use of effective bandwidth eb as $A_h[k] = C - eb$. The former gives a strictly conservative estimate of the available bandwidth on the link for the entire duration. The latter gives a more realistic estimate which is tunable based on the network operators bandwidth requirements. Effective bandwidth [108] is a measure of the traffic stream that characterizes its steady state behavior and is given as

$$eb(s) = \lim_{t \to \infty} \frac{1}{st} \log E[e^{sL[0,t]}] \tag{36}$$

where s is the decay rate of queue size distribution tail probability and $L[0,t]$ is the total traffic load arrived during the time interval $[0,t]$. The equation (36) provides an effective bandwidth value between the peak and average traffic in $[0,t]$. An on-line block estimator for the effective bandwidth formulation is given in [109] and can be modified as given in Figure 21.

After obtaining the actual load $[L_\tau[k+1], \ldots, L_\tau[k+h]]^T$ at time $(k+h)t$, the prediction error vector $[e_\tau[k+1], \ldots, e_\tau[k+h]]^T$ can be found with elements as:

$$e_\tau[k+a] = \left(L_\tau[k+a] - \widehat{L}_\tau[k+a]\right)^2 \quad \text{for } a = 1, \ldots, h.$$

Next, an algorithm is proposed to estimate new values for p and h based on a metric derived from the mean (ν) and standard deviation (σ) of error e_τ. The algorithm is given in

```
Algorithm:
  1. Initialize $M = 0$ and $i = k$,
  2. Obtain the prediction $\hat{L}_\tau[i]$,
  3. Update $M = (1 - \frac{1}{i})M + \frac{1}{i}exp(s\tau\hat{L}_\tau[i])$,
  4. If $i < k + p$, go to step 2,
  5. $eb(s) = \frac{1}{s\tau}log(M); Stop.$
```

Figure 21: Effective Bandwidth Estimator Algorithm.

```
Algorithm:
  1. If $\sigma/\nu > Th_1$, decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ multiplicatively.
  2. If $Th_1 > \sigma/\nu > Th_2$, decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ additively.
  3. If $\sigma/\nu < Th_2$, then:
     (a) If $\nu > Th_3 * M_E^2$, decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ additively.
     (b) If $Th_3 * M_E^2 > \nu > Th_4 * M_E^2$, keep $h$ and $p$ constant.
     (c) If $\nu < Th_4 * M_E^2$, increase $h$ and decrease $p$ till $p_{min}$ additively.
```

Figure 22: Algorithm for h and p Determination.

Figure 22. In the algorithm, M_E is the maximum error value and h_{min} and p_{max} have been introduced because small values of h imply frequent re-computation of the regression coefficients and large values of p increase the computational cost of the regression. Also, the thresholds Th_1 to Th_4 help to decide when to change the values of the parameters p and h. They are determined based on the traffic characteristics and the conservatism requirements of the network domain. They represent the confidence in the estimation procedure in terms of prediction errors.

5.4 Non-periodic Single-step Prediction Based Algorithm

The ABEst algorithm in the previous section uses samples obtained with constant frequency for the linear regression to obtain the prediction for the future utilization. This method is computationally less intensive as the prediction coefficients can be calculated by solving a Toeplitz matrix. However, this method suffers from the drawback of reduced efficiency as multi-step prediction is error-prone. Thus, in this section, a link utilization estimation algorithm is presented that is based on single step prediction. To provide more information for the prediction, the frequency of the samples used in the prediction is varied.

5.4.1 Problem Formulation

The following are defined for a link between two nodes i and j:

- C: Capacity of link in bits per sec,
- $A(t)$: Available capacity at time *t* in bits per sec ,
- $L(t)$: Traffic load at time *t* in bits per sec,
- τ_{min}: Length of the minimum sampling interval of MRTG,
- τ: Length of the current sampling interval of MRTG (integral multiple of τ_{min}),
- $\eta, \eta \in \mathbf{N}$: τ/τ_{min}, the number of τ_{min} in current τ,
- $L_\tau[k], k \in \mathbf{N}$: Average load in $[(k-\eta)\tau_{\text{min}}, k\tau_{\text{min}}]$,
- p is the number of past measurements in prediction,
- $\widehat{A}[k+\eta]$ is the prediction for $A(t)$ for time $[k\tau_{\text{min}}, (k+\eta)\tau_{\text{min}}]$.
- $\widehat{L}_\tau[k+\eta]$ is the prediction of utilization for time $[k\tau_{\text{min}}, (k+\eta)\tau_{\text{min}}]$.

A linear regression based algorithm is proposed for prediction of the link utilization. Superimposed on the regression is another algorithm to vary the sampling frequency on a scale smaller than the sampling frequency itself. During the interval when the sampling frequency is held constant, the linear prediction can be specified as

$$L_\tau[k+\eta] = \sum_{n=0}^{p-1} L_\tau[k-\eta n]\, a_p[n] \tag{37}$$

where on the right side are the past samples $L_\tau[k-\eta n]$ and the prediction coefficients $a_p[n]$ and on the left side, the predicted value. A Gradient Adaptive Lattice filter [107] is used to find the prediction coefficients to minimize the forward and backward prediction errors. The value of η can be dynamically changed based on the traffic dynamics to calculate a more efficient prediction of the utilization.

5.4.2 Available Bandwidth Estimation Algorithm

The Method for Available Bandwidth Estimation (MABE) algorithm is given in Figure 23. In step 2 of the algorithm, the reflection coefficients of the Gradient Adaptive Lattice filter need to be found. The lattice filter minimizes the forward and backward prediction errors, $f_j(k)$ and $b_j(k)$ respectively, for $j = 1, 2, \ldots, p$, where

$$\begin{aligned} f_j(k) &= L_\tau[k] + \sum_{n=1}^{p} L_\tau[k-\eta n]\, a_j[n], \\ b_j(k) &= L_\tau[k-\eta j] + \sum_{n=1}^{p} L_\tau[k-\eta j+\eta n]\, a_j[n]. \end{aligned}$$

With the steepest descent approach to minimize the total error, the reflection coefficient update equation for $j = 1, 2, \ldots p$ is

$$\Gamma_j(k+1) = \Gamma_j(k) - \rho_j(n)\left\{f_j(k)b_{j-1}(k-1) + f_{j-1}(k)b_j(k)\right\}$$

where ρ_j is a time-varying step size used to normalize the gradient adaptive lattice filter

$$\rho_j(n) = \frac{2}{\sum_{l=0}^{n}(0.5)^{(n-l)}(f_{j-1}^2(l) + b_{j-1}^2(l-1))}.$$

From the reflection coefficients, the prediction coefficients a_p can be calculated as

$$a_{j+1}[i] = a_j[i] + \Gamma_{j+1}a_j[j-i+1]$$

where j is the order of the prediction and $i = 0, 1, \ldots, j+1$. The prediction coefficients a_p can then be used to predict $\widehat{L}_\tau[k+\eta]$ using Eq. 37. Next step is to obtain an estimate of the available bandwidth $\widehat{A}[k+\eta]$ for the interval $[k\tau_{\text{min}}, (k+\eta)\tau_{\text{min}}]$ as $\widehat{A}[k+\eta] = C - \widehat{L}_\tau[k+\eta]$.

```
MABE Algorithm:
 1. At time instant $k$, available bandwidth measurement is desired.
 2. Find the coefficient vector $a_p$ using Gradient Adaptive Lattice method for given $p$ and the previous measurements.
 3. Predict $\hat{L}_\tau[k+\eta]$ from Eq. 37 and $[L_\tau[k-(p-1)\eta], \dots, L_\tau[k]]$ .
 4. Find $\hat{A}[k+\eta]$.
 5. At time $(k+\eta)\tau_{\text{min}}$, get $L_\tau[k+\eta]$.
 6. Find the error $e_\tau[k+\eta]$ and its mean $(\nu)$ and standard deviation $(\sigma)$.
 7. Set $k = k+\eta$.
 8. Obtain new value for $\eta$ from Fig. 2.
 9. If $\eta$ has changed then call the transient algorithm.
10. Go to step 1.
```

Figure 23: The MABE Algorithm.

```
Algorithm:
 1. If $\sigma/\nu > Th_1$, decrease $\eta$ multiplicatively till $\eta_{\text{min}}$ i.e. $\eta = \eta/N, N > 1$.
 2. If $Th_1 > \sigma/\nu > Th_2$, keep $\eta$ constant.
 3. If $\sigma/\nu < Th_2$, increase $\eta$ multiplicatively till $\eta_{\text{max}}$ i.e. $\eta = N\eta$.
```

Figure 24: Algorithm for η.

After obtaining the actual load value $L_\tau[k+\eta]$ at time $(k+\eta)\tau_{\text{min}}$, the prediction error vector $[e_\tau[k-(p-2)\eta], \dots, e_\tau[k+\eta]]^T$ can be calculated. The elements of the error vector are given as

$$e_\tau[k+a\eta] = \left(L_\tau[k+a\eta] - \widehat{L}_\tau[k+a\eta]\right)^2, \quad a = -p+2, -p+1, \dots, 1.$$

Next, an algorithm is needed to estimate new value for η based on a metric derived from the mean (ν) and standard deviation (σ) of error e_τ. The algorithm is given in Figure 24.

In the algorithm, η_{min} and η_{max} have been introduced because small value of η implies frequent regression re-computation while large value of η decreases the reliability of the regression. Also, the thresholds Th_1 and Th_2 help to decide when to change the value of η. They are determined based on the traffic characteristics and the conservatism requirements of the network domain. They should be chosen such that the variations in η are not too frequent. If the algorithm of Figure 24 decides to change the value of η, *i.e.* the sampling frequency is changed, a transition algorithm can be used, as explained next. A decrease in the value of η implies more frequent sampling in the future. During the transition stage, since enough samples from the past (with proper spacing) are not available for the prediction, some predictions can be obtained with the old far-spaced samples and then linear interpolation between them can provide closely spaced samples. On the other hand, if η is increased, the sampling interval needs to be increased. This can be achieved by obtaining some predictions with the old near-spaced samples and then filtering to drop some of the obtained values. This algorithm introduces a smooth transition period where still enough samples are available for the linear regression. Once the value of η remains constant for a while, the system is able to achieve a stable operating point where the required past samples can be obtained directly from measurements.

5.5 Implementation

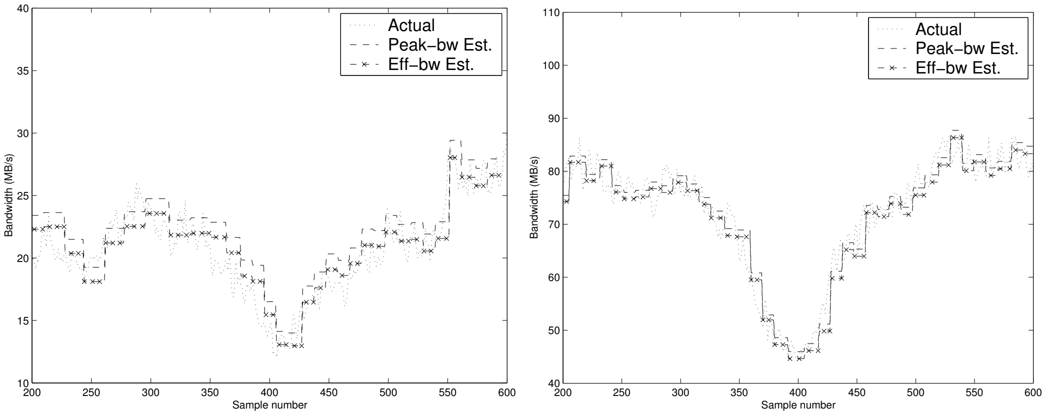
In an SNMP network, the managed devices collect and store management information in MIBs and make it available to the managers through an agent running on the device. Each element in the MIB is identified by a sequence of numbers called Object Identifier (OID). The NMS can then retrieve specific information from the device using these identifiers. IETF has defined a standard [105] with specifications, grouping and relationships of managed objects in an SNMP compatible network. MRTG can be used to sample rates of almost any OID. By default, it is used to periodically fetch in-bound and out-bound traffic counters on the router interfaces and calculate the traffic rate on each one of them. These variables are available through the OIDs corresponding to in-bound and out-bound counters (in bytes) for each interface. MRTG stores the traffic rates for each interval of time, calculated by taking the difference of the counters and dividing by the interval length. MRTG database has a very simple layout. Each line has 5 values: time-stamp, in-bound average rate, out-bound average rate, in-bound maximum rate and out-bound maximum rate. MRTG also keeps track of the counter values at the last sample in order to calculate the rates for the next period.

Even though MRTG provides real-time available bandwidth measurements for a link, it may not be useful because of the 5 minute averaging intervals. Even if the RRDTool is used, the 300 seconds interval is hard coded in the MRTG source code. Patches are available to bring the interval detail down to 1 minute. However, in some cases, 1 minute might still be too coarse. Thus, MRTG++ was developed as a patch to MRTG. MRTG++ provides up to 10 seconds detail which is a much finer granularity of measurements. First of all, the Round Robin Database (RRD) must be created with enough slots to store the larger amount of information. Then, the consolidation function parameters, i.e. how many samples the database will consider when calculating the average, must be adjusted for the new intervals. The current database in TEAM is able to store 10 seconds averages for up to 24 hours. Next step is to modify the script to send the correct queries to RRDTool when creating graphs. Since the intervals have changed, the scale and the set of data for the script must also be changed. Finally, MRTG++ must be run every 10 seconds to get the information from the routers.

The NMS should decide the optimal MRTG measurement period based on the traffic characteristics, the required granularity for the measured values and the appropriate time-scale of the application utilizing the measured values.

5.6 Performance Evaluation

The proposed algorithm ABEst for available bandwidth estimation on a link does not make any assumption about the traffic models. It works based on the measurements obtained



(a) Atlanta Router

(b) Cleveland Router

Figure 25: Input Traffic.

from the network link. Thus, there is no need to use a network simulator. Instead, the algorithm can be applied to traffic traces obtained from real networks. The following results present a combination of the actual traffic profile and the traffic prediction by ABEst. The predicted available bandwidth is not presented because that can be calculated by taking the difference of the link capacity and the utilization and thus it does not present significant information, when compared to the predicted utilization.

The choice of the thresholds Th_1, Th_2, etc. and h_{min}, p_{max} used for updating the values of p and h in Section 5.3.2 has to be made by the network manager depending on the conservativeness requirements of the network operation. The following results have been obtained by choosing $Th_1 = 0.9$, $Th_2 = 0.7$, $Th_3 = 0.5$, $Th_4 = 0.3$ and $h_{min} = 10$, $p_{max} = 50$. All the traffic traces used in the following results have been obtained from Abilene [110], the advanced backbone network of the Internet2 community, on March 13, 2002. In Figure 25(a), the ABEst algorithm is applied to the input traffic on the Atlanta router of the Atlanta-Washington D.C. link. In Figure 25(b), same is done for the input traffic on the Cleveland router from the Cleveland-NYC link. In both cases, the first curve shows the actual traffic profile and the other two curves show the prediction by utilizing ABEst. The first of the two utilizes the peak-based estimation whereas the second utilizes

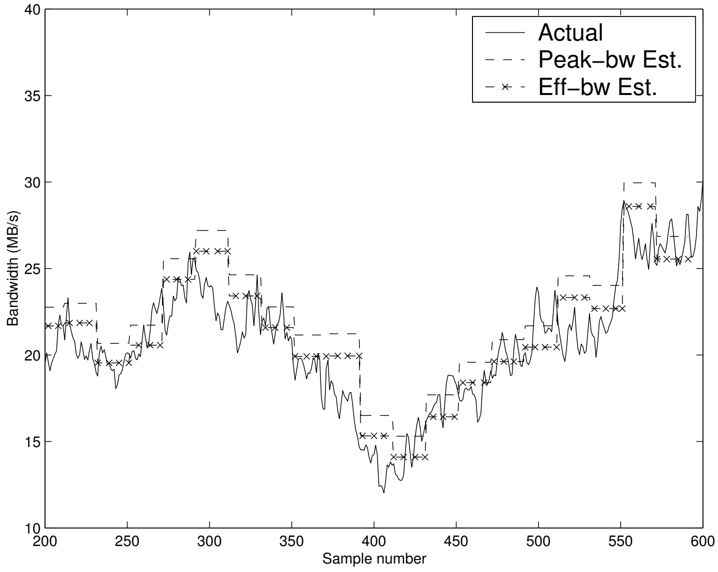


Figure 26: Input Traffic on Atlanta Router (h_{min} = 20).

the effective bandwidth-based estimation. As can be seen, in both cases, the utilization estimation obtained by taking the peak prediction provides a conservative estimate, whereas the estimation using the effective bandwidth provides an estimate for lower resource utilization. Also, when h_{min} is increased, the estimation becomes worse (see Figure 26), in the sense that it does not follow the sequence closely but is still very conservative. Overestimation can be used as a metric to quantitatively measure the performance of the proposed scheme ABEst. For the case in Figure 25(a), the mean overestimation is 1.31 MB/s for the peak estimation procedure whereas it is 0.73 MB/s for the effective bandwidth estimation procedure. Similar values are obtained for the case depicted in Figure 25(b).

When compared with MRTG, ABEst provides the available bandwidth estimates less frequently without a large compromise in the reliability of the estimate. In other words, the utilization profile obtained as a result of MRTG coincides with the actual traffic profile in Figure 25(a) and (b), but ABEst provides an estimate of the link utilization which is nearly accurate with a reduced computational effort.

Chapter 6

End-to-end Available Bandwidth Measurement

With the growing traffic in the DiffServ/MPLS domain, tools are needed to understand the composition and dynamics of the traffic. In this chapter, a tool for measurement of end-to-end available bandwidth between a node pair is presented. The tool combines the advantages of both active and passive measurement methodologies to obtain accurate, reliable measurements of the available bandwidth along a path. The tool utilizes the interface information from the MIBs in the routers along the path. The functionality of the tool is distributed between both the source and destination of the path whose measurement is desired. The source sends measurement packets that collect information along the path and are returned back by the destination to the source. This measurement tool was introduced in [111].

This chapter is organized as follows: The motivation for the development of an end-to-end available bandwidth measurement algorithm is given in Section 6.1. In Section 6.2, other end-to-end measurement algorithms are presented. Then, in Section 6.3, the description of the tool is presented along with the probe packet structure, hop functionalities etc. Finally, the performance evaluation for the tool is presented in Section 6.4.

6.1 Motivation

As detailed in Section 5.1, measurement is necessary for a network, from both the user and service provider's point of view. Common users can only measure the end-to-end metrics. The metrics with local significance at each router can only be measured by the network operators. The approaches to monitor a network are *active* or *passive*. First gives a measure of the performance of the network whereas the latter of the workload on the network.

Available bandwidth (together with other metrics like latency, loss etc.) can predict the performance of the network. The *available bandwidth* of a link is the maximum throughput provided to a flow despite the current cross-traffic, when contrasted with the *capacity* which is the maximum throughput provided to a flow in absence of cross-traffic. Based on the bandwidth available in various segments of the network, the network operator can obtain information about the congestion in the network, perform the admission control, routing, capacity provisioning etc. The available bandwidth can be measured for individual links of the network. The end-to-end available bandwidth information can be obtained by a concatenation of the available bandwidth measurements of the individual links comprising the path. However, this approach can be very inefficient as the amount of data collected grows as the path size increases and a central data analysis station will be required. Thus, tools have to be devised that can measure the end-to-end available bandwidth directly and accurately from the path. The end-to-end available bandwidth information can be used for selection of alternative paths, selection of web servers etc.

In this chapter, a tool is proposed for measuring the end-to-end available bandwidth over a path that can possibly span across multiple domains. The tool is needed to answer the question "Where in the path between the two endpoints is the least bandwidth available to a flow and how much is it?". Currently this is hard to do because the available bandwidth, even for a single link, shows large variations with time, the path may change during the measurement, etc.

The need for an active network measurement tool is well established due to the accuracy requirements. The proposed tool is efficient, easy to implement, and a combination of active and passive approaches. This way, it derives the benefits of both the measurement approaches. The tool is designed such that the measurement packets are processed with about the same computation level as IP forwarding.

6.2 Related Work

The available bandwidth of a link is indicative of the amount of load that can be routed on the link. Obtaining an accurate measurement of the available bandwidth can be crucial to effective deployment of QoS services in a network. Available bandwidth can be measured using both active and passive approaches. Various tools and products are available that can be used to measure bandwidth of a path in the network. The first tool that attempted to measure available bandwidth was cprobe [56]. This tool estimated the available bandwidth based on the dispersion of long packet trains at the receiver. A similar approach was given in pipechar [57]. The underlying assumption for these tools is that the dispersion of long packet trains is inversely proportional to the available bandwidth. However, this is not true [58]. Another measurement technique, Delphi [59], assumes that the path can be well modeled by a single queue and so it is not applicable when there are significant queuing delays in several links of the path. In [46], the authors have described a few bandwidth estimation algorithms. They can be split into two families: those based on pathchar algorithm and those based on Packet Pair algorithm. In the pathchar approach, packets of varying sizes are sent with increasing values of the Time-To-Live (TTL). The packet pair algorithm measures the bandwidth of the narrow link of a route. It operates by sending two packets which get queued along the narrow link of the path and their time-spacing provides estimate of the narrow link bandwidth. In [49], the authors have proposed another tool to measure narrow link bandwidth based on packet pair technique. Some other tools based on the same technique for measuring bottleneck bandwidth (of narrow link) of a route have been proposed in [50, 51]. In [60], a tool to measure the available bandwidth of a path is presented. It is an active approach based on transmission of self-loading periodic measurement streams. This scheme sends traffic at increasing rates from the source to the destination until the rate finally reaches the available bandwidth of the tight link after which the packets start experiencing increasing amounts of delay. Thus this scheme can be highly intrusive even though momentarily. MRTG is a tool, based on SNMP, that gives periodic measurements

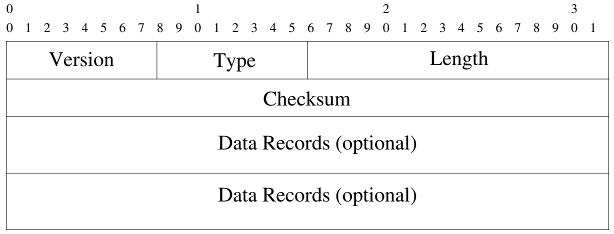


Figure 27: Measurement Packet Format.

of the utilization of a particular link along the path. To obtain statistics of a link via MRTG, the SNMP query needs access to the router. Also, MRTG obtains available bandwidth estimates over periods of length 5 minutes.

Most of the tools/approaches described above obtain estimates of the capacity of the path, rather than the available bandwidth. Even the ones that do measure available bandwidth operate under a lot of assumptions about the packet pair and their queuing along the path. In the following sections, a tool is presented for the estimation of available bandwidth along a path which is accurate, scalable and flexible.

6.3 Measurement Tool

In this section, the proposed Tool for End-to-end Measurement of Available Bandwidth (TEMB) is described. The tool is needed to answer the question "Where in the path between the two endpoints is the least bandwidth available to a flow and how much is it?". Currently this is hard to do because the available bandwidth, even for a single link, shows large variations with time, the path may change during the measurement, etc. TEMB utilizes the interface information from the Management Information Bases (MIBs) in the routers along the path. The functionality of TEMB is distributed between both the source and destination of the path whose measurement is desired. The source sends measurement packets that collect information along the path and are returned back by the destination to the source.

6.3.1 Packet Structure

The TEMB tool is based on the use of measurement packets to probe the available bandwidth along the path. The format of the measurement packet is shown in Figure 27. In the packet, the various fields are:

- Version: Set to 0,
- Type: Set to 0 if the packet is sent from the source to the destination and is routed hop-by-hop by the network, 1 if the path from the source to the destination is already pinned and encoded into the packet, and 2 if the packet is being returned from the destination, as explained later,
- Length: Total length of the TEMB packet in bytes,
- Checksum: CRC for the whole packet,
- Data Records: Modified by each hop, as explained later.

Assuming that the TEMB packets can encounter links with the smallest MTU (576 bytes), TEMB is designed such that the measurement packets can not exceed a size of 556 bytes, accounting for the 20 byte IP overhead. The 576 byte limit is imposed as longer packets might get fragmented and eventually discarded. The 556 byte payload implies that a maximum of 34 data records (16 bytes each) can be gathered by a TEMB packet, which is a reasonable limit to the number of hops encountered between any source-destination pair across the world. The TEMB measurement packets are encapsulated into IP packets as explained later.

6.3.2 Destination Functionality

When a measurement packet finally reaches its destination, it has gathered information from TEMB-compatible hops in the path from the source to the destination, in the form of the data records in the packet. Since the information is along the path and is unidirectional,

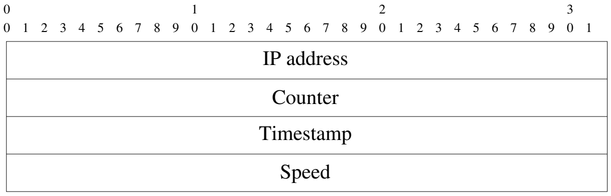


Figure 28: Measurement Packet Data Record.

the source of the path is better suited to analyze the information from the measurements. Thus, the measurement packet has to be returned to the source. Towards this end, the destination interchanges the source and destination fields of the IP header of the measurement packet, changes the type field to 2 to indicate a packet back from the destination to the source and sends the packet to its queues for transmission.

6.3.3 Hop Functionality

Each hop on the path from the source to the destination appends its information to the measurement packets, if it is TEMB-compatible. The information is in the form of data records which are included at the end of the measurement packet. The structure of the data record is shown in Figure 28. Each data record contains the IP address of the out-bound interface of the router, the counter for the number of octets that have passed that interface at the processing time, the time-stamp at which the packet was processed by the router, and the speed of the outgoing interface. The value of the counter can be obtained by looking at the ifOutOctets object in the interfaces group of the MIB-II in the router. As the router has modified the measurement packet, it has to recompute the length of the packet and the CRC checksum for the modified packet. These values then have to be substituted into the packet and the packet then queued for transmission downstream.

6.3.4 Source Functionality

In the design of TEMB, most of the operational burden is given to the source router of the path. This is because it is closest to the user who demands the bandwidth and the QoS and can inform the user about the path conditions. The source of the path has to assemble the initial measurement packet with the initial data records and correct packet length and checksum. Then, the measurement packet is encapsulated into an IP packet and the appropriate link layer packet. The source makes multiple copies of this packet and sends them over the path to the destination. Multiple packets are sent to obtain a correct estimate of the identity of the tight link and its available bandwidth. Also the source node has to analyze the incoming packets and take further measures to obtain more refined available bandwidth measurements. The detailed operation of the tool is described next.

6.3.5 Overall Operation

TEMB is a tool designed to measure the available bandwidth along a path between a source and a destination. The operation of TEMB can be split into two parts. The first obtains crude estimates of available bandwidth along all the links of the path and determines the least amount among them as a method to identify the tight link along the path. The second part obtains more accurate measurement of the available bandwidth on the identified tight link. The tool operates by sending the measurement packets from the source to the destination. TEMB is designed to initially transmit 10 measurement packets during an interval of 1 sec. The number 10 was chosen because it gives a reasonable approximation of the bandwidths along the path without being highly intrusive. These packets gather information along the path and the destination sends the packets back to the source. If the traffic profile along the path is highly variable, TEMB is designed to dispatch another set of 10 packets in one second to obtain better identification of the tight link of the path.

Suppose N out of the initial 10 packets are finally back at the source for analysis. Let T denote the set of all the time-stamps gathered by the packets, *i.e.*, $T = \{t_1, t_2, \ldots, t_N\}$, where the elements have been arranged in increasing order, *i.e.*, $t_1 < t_2 < \ldots < t_N$. Let P denote the list of successive interfaces encountered by the measurement packets, *i.e.*, $P = \{I_1, I_2, \ldots, I_h\}$ where h is the number of hops in the path. Let C_I denote the set of counters for interface I in the path, *i.e.*, $C_I = \{c_{I1}, c_{I2}, \ldots, c_{IN}\}$, where c_{Ik} denotes the counter for interface I along the path at time t_k. Since the set T is arranged in an increasing order, the elements of C_I are also monotonous non-decreasing as the number of packets crossing an interface is always non-negative. Let S_I be the speed of the interface, as gathered by the measurement packets. Then, the utilization of the interface is calculated from the k^{th} sample as

$$U_{Ik} = \frac{c_{Ik} - c_{I(k-1)}}{t_k - t_{(k-1)}} \quad \text{for } k = 2, 3, \ldots, N$$

and the available bandwidth is $A_{Ik} = S_I - U_{Ik}$. Once these $(N - 1)$ estimates are obtained, TEMB tries to identify the tight link. If all the estimates agree on a certain interface being the one with the minimum available bandwidth and the estimated values are similar, TEMB identifies the tight link. On the other hand, if the estimates disagree on either the interface or its available bandwidth, TEMB sends the next batch of measurement packets. The agreement about the identity of the tight link is reached if at least a certain percentage $(agree_{link})$ of the estimates concur. The agreement about the estimated value of the available bandwidth is reached if all the $(N - 1)$ estimates for the interface I are less than a a certain percentage $(agree_{avail})$ of the minimum estimate. Note that the values of both $agree_{link}$ and $agree_{avail}$ should be very close to 100 but the former should be less than 100 and the latter greater than 100.

Finally the average available bandwidth of the selected interface I is

$$A_I = \frac{1}{n * (N-1)} \sum_{k=1}^{p*(N-1)} A_{Ik} \tag{38}$$

where n is the number of attempts that TEMB made during the first step. The identified link is then chosen for further investigation. Also, if the estimated available bandwidth for any other interface falls within 150% of the lowest value, that interface is also marked critical and qualifies for further investigation. This is done due to the non-stationary nature of cross-traffic.

The tool is designed to operate in two cases: a. When the path between the source and destination is the min-hop path, b. When the available bandwidth measurement is required for a non-min-hop path between the source and destination. In the first case, the measurement packets formed at the source have empty data records, the type is set to 0, and the measurement packets are IP encapsulated. As they move down the path, each hop adds its data record and forwards the packets to the destination by utilizing the pre-existing IP lookup tables. In the second case, the data records are already included in the measurement packets at the source. The data records contain the IP address of the hops along the path. In the measurement packets, the type value is set to 1. The hops of the path modify their data records by including the interface information and then queue the packets for transmission towards the next hop, as recorded in the next data record. In this way, the available bandwidth measurements can be obtained for predetermined paths.

Once the identification of the tight links of the path is done, a more accurate estimation of its available bandwidth is desired. This is done by utilizing an MRTG based approach that is passive in nature and has been monitoring the interface over time. This approach is similar to the available bandwidth estimation algorithm in Chapter 5.

6.4 Performance Evaluation

In this section, the results of the experiments and simulations to verify the operation of the proposed tool TEMB are presented. The simulations are divided into two categories to verify the two parts of the functionality of TEMB. First, some implementation details about TEMB are discussed.

6.4.1 Implementation Details

The proposed tool TEMB for identification of the tight link along a path and subsequent available bandwidth measurement along the tight link is designed to be as much non-intrusive as possible. Unlike some other schemes that transmit packets at speed higher than the the tight link available bandwidth to get an estimate, this tool sends 10 packets in a second per path for each measurement required, which may be repeated a couple of times.

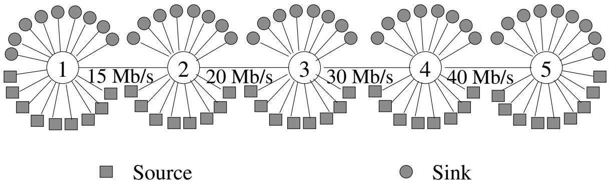


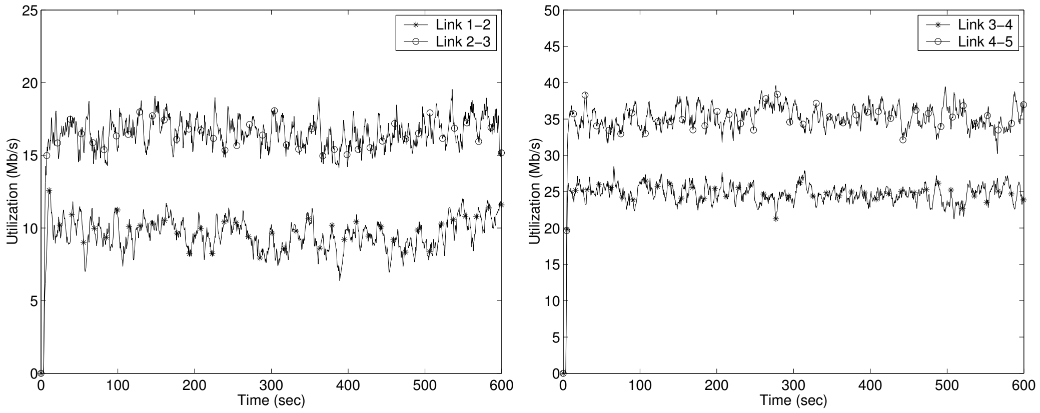
Figure 29: TEMB Simulation Topology.

Thus, it combines the advantages of both the active and passive approaches of measurement by using a hybrid tool.

One advantage of using TEMB is that the router time-stamps saved in the measurement packets have only local significance, *i.e.*, they are not correlated with other routers. The routers do not need information about the real time reference of their time-stamp. Only the difference in the consecutive time-stamps is used to calculate the utilization for that interval. Also, the ordering of the measurement packets does not matter when they reach the destination.

6.4.2 Simulator Description

The well-known network simulator ns is used to simulate the topology of Figure 29. Traffic is generated by utilizing various UDP and TCP sources which are attached to different nodes in the path. The UDP senders are ON/OFF sources. The durations of the ON and OFF periods are selected from a Pareto distribution with parameter β. During the ON period, the transmission rate of the sender is a constant configured rate. In the simulations, three different UDP sender profiles have been used. A combination of Pareto-based ON/OFF sources is used as it leads to long-range dependence in the multiplexed traffic. Sources send traffic to sinks located at all the nodes in the network. As a combination of ON/OFF and TCP sources with different source/sink pairs and average sending rates is used, the resultant traffic on the links of the simulation topology is not correlated and is a good representative for current Internet traffic. For verification of the MRTG-based tool, traffic measurements from a real Internet backbone are used.



(a) Link 1-2 and 2-3

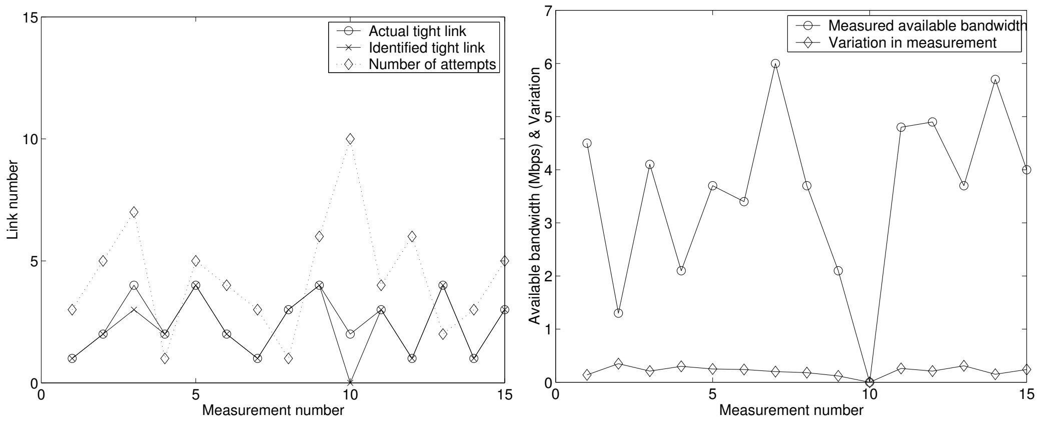
(b) Link 3-4 and 4-5

Figure 30: Utilization of the Four Links.

6.4.3 Path Probing Results

The goal is to demonstrate how well the path probing tool of TEMB works over time when presented with different traffic patterns on links of the path and how the parameter tuning affects the performance. The simulation topology consists of 5 nodes arranged in a totem-pole (Figure 29). While this topology does not cover the full heterogeneity of the Internet, it is sufficient for this study since it provides different, uncorrelated traffic patterns on the different links. The end-to-end available bandwidth measurement is desired from node 1 to node 5. A combination of TCP and UDP sources are attached to the nodes. The value of $agree_{link}$ is set at 85% and $agree_{avail}$ at 120%. Thus, at least 85% of the measurements have to point to a certain link for it to be identified as the tight link of the path and the maximum measurement obtained for that interface should not exceed 120% of the minimum. This constraint is applied to ensure that the traffic profile does not vary a lot during the measurement to guarantee a good measurement estimate for the available bandwidth of the tight link. Figure 30 shows a sample utilization profile for the four links, obtained from one simulation run. If the path probing mechanism of Section 6.3 is applied to probe the available bandwidth at time instant 360 sec, it is obtained with 77% agreement that the least available bandwidth is along the link 1-2. This is because 7 of the 9 measurements pointed towards link 1-2. If the application that desires the measurement demands a higher value of $agree_{link}$, then the next batch of measurement packets is sent at time 362 sec. From this batch, all 9 measurements point towards link 1-2. Thus the confidence in identification of link 1-2 as the tight link becomes 89% which is higher than the threshold $agree_{link}$. It is also observed that the maximum among the measurements obtained for the link 1-2 is about 113% of the minimum, which is below the limit set by $agree_{avail}$. Thus the tight link is identified as link 1-2 and its available bandwidth is the minimum measurement obtained, *i.e.* 2.6Mbps. Also, link 4-5 is marked as critical as its available bandwidth measurements fall below the 150% mark of the link 1-2 minimum 2.6Mbps. If, on the other hand, the path probing mechanism was applied at time 390 sec, it would be seen that 7 of the 9 measurements pointed towards link 2-3. As this agreement (77%) is below $agree_{link}$, the next 10 measurement packets are sent at time 393 sec. Among them, 88% point towards link 4-5, which necessitates another attempt at tight link identification. The third attempt at time 395 sec returns link 2-3 with an agreement of 88% which leads to an overall agreement of 55% for link 2-3. This leads to 6 further attempts which all point to link 2-3 with a 100% agreement before the link 2-3 is finally chosen as the tight link. These two scenarios illustrate that the computational effort and the intrusiveness of the scheme is highly dependent on the parameters $agree_{link}$ and $agree_{avail}$ specified by the application. If the application needs a high level of agreement before identifying the tight link, multiple attempts may be necessary to achieve the same.

In Figure 31(a), the results for 15 independent runs of the path probing mechanism for the topology and setup in Figure 29 are shown. The link 1 is the link between nodes 1 and 2, link 2 between nodes 2 and 3, and so on. The values of $agree_{link}$ and $agree_{avail}$ were set to 80% and 140%, respectively. As can be seen, the mechanism only falters in one case (measurement 3). In the case of measurement experiment 10, 10 attempts were made unsuccessfully to determine the tight link, whereupon the particular experiment was deserted. Also shown in the figure with the dotted line is the number of attempts that the



(a) Verification of path probing mechanism.

(b) Variation of available bandwidth.

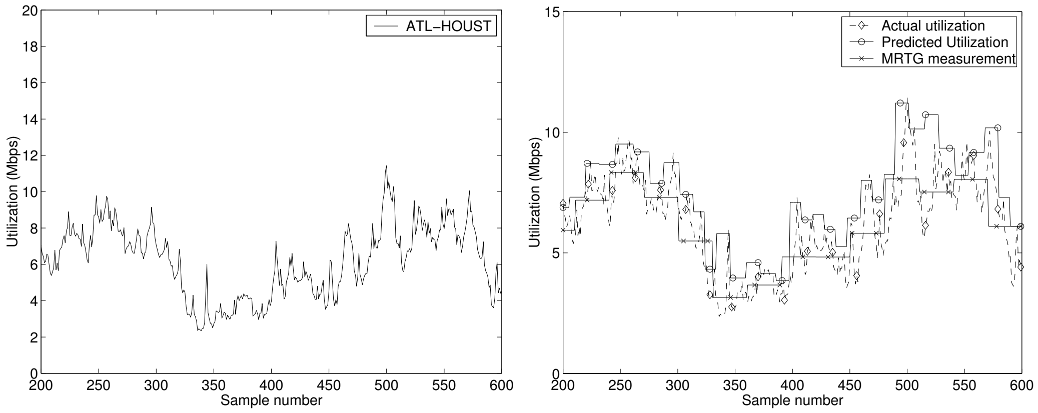
Figure 31: TEMB Performance.

path probing tool made at the measurement before arriving at the final result. As can be seen, the number of attempts is not very high, demonstrating that the scheme is not highly-intrusive (as each attempt includes transmitting 10 packets in 1 sec). In Fig. 31(b), the variation of the measured available bandwidth of the identified tight link for the same 15 experiments is shown. The variation is defined as the ratio of the difference of the maximum measurement and the minimum measurement to the minimum measurement. Also shown is the final estimate of the tight link available bandwidth. The figure gives a slight hint that the variation in the available bandwidth increases as the tight link utilization increases.

6.4.4 MRTG Based Approach Results

The MRTG-based tool is designed to find, for the link identified by the path probing mechanism, a more accurate estimate of the available bandwidth. It operates, independent of any assumptions about the traffic models, based on the actual measurements obtained from the link. To validate the performance of the MRTG-based tool, traffic traces are obtained from measurements available from real Internet backbones. This is done because it gives insight into the performance of the tool for observed real traffic traces.

The choice of the parameters used in the tool (Th_1, Th_2, etc. and h_{min}, p_{max}) for



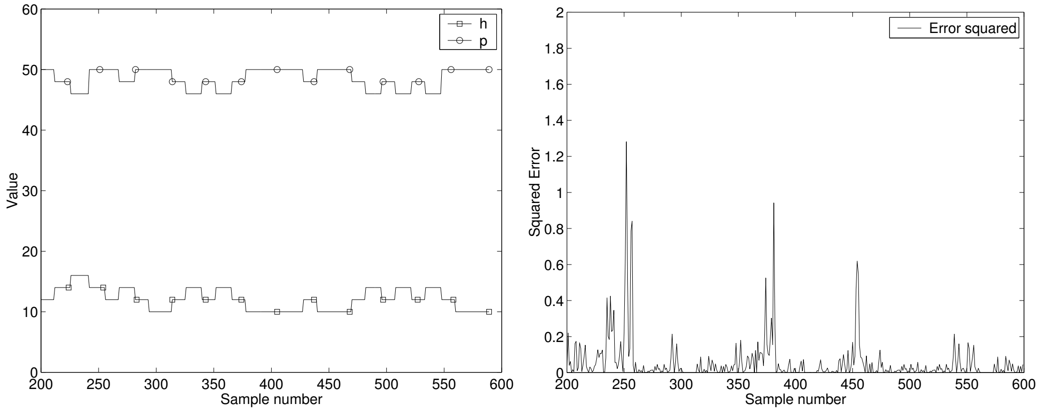
(a) Out Traffic from ATL on ATL-HOUST.

(b) Predicted Utilization.

Figure 32: MRTG Based Tool's Performance for Real Traffic Trace.

updating the values of p and h has to be made by the network operator depending on the conservativeness requirements of the network operation. The following results were obtained by choosing $Th_1 = 1.1$, $Th_2 = 0.9$, $Th_3 = 0.7$, $Th_4 = 0.5$ and $h_{\text{min}} = 10$, $p_{max} = 50$. Also, the representative utilization for an interval is fixed at the maximum predicted utilization, in order to provide a very conservative estimate for the link available bandwidth. The traffic traces have been obtained from Abilene, the advanced backbone network of the Internet2 community of universities, on July 1, 2002. The performance was checked for traffic traces obtained from various links of the network, but the following results are for the traffic trace between Atlanta and Houston routers, for the outgoing traffic from the Atlanta router (shown in Figure 32(a)). In the figure, the samples are collected with a time granularity of 10 seconds.

When the tool is applied to this traffic trace, the predicted utilization is shown in Fig. 32(b). Also shown in the figure is the utilization profile that would be observed if MRTG was applied to the trace, with its 5 minute averaging. As can be seen, the proposed tool performs much better than MRTG as it gives a very conservative utilization estimation. Also combined with the tool is the capability to predict, for a future small interval, the utilization with a high degree of confidence. In this figure, the available bandwidth profile is



(a) Variation in Parameters h and p.

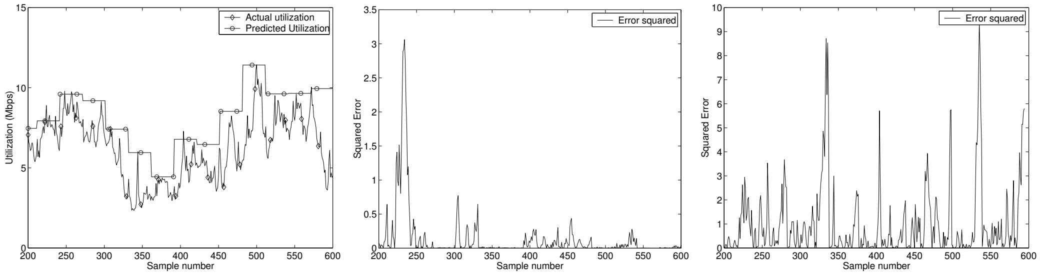
(b) Error in Prediction.

Figure 33: Prediction Performance.

not shown as it can be obtained simply by subtracting the utilization from the total capacity of the link. Also, the predicted utilization shows more relevant information when placed against the actual measurements. In the figure, the results are shown from sample number 200 onwards as the initial samples are used to stabilize the tool. In Fig. 33(a), the values that were assigned to the forecast parameters p and h during the experiment are shown. As can be seen, the values of h and p are not always the limits set by h_{min} and p_{max}. This shows that the scheme was able to gain confidence in its prediction for certain intervals. Next, in Fig. 33(b), the squared error of the prediction is shown. As can be noticed from comparing the figure with Fig. 32(b), the error does not exceed 10% of the actual utilization.

Next, in Figure 34(a), the value of h_{min} has been increased to 30 forcing the tool to predict for longer intervals even though the confidence in the prediction may not be so high. As can be seen, the prediction error increases (as expected) but the prediction is still very conservative.

To validate the operation of the proposed tool, the scenario where the parameters h and p are fixed are depicted in Figs. 34(b) and 34(c), respectively. In other words, for the case in Fig. 34(b), h was fixed at 30 to obtain results for periods equal to MRTG intervals of 5 min. For the case in Fig. 34(c), p was fixed at 30 to obtain the results. This value was picked as it



(a) Increased h_{min} (b) Constant h (c) Constant p

Figure 34: Effect of Parameter Variation.

is the average of the range allowed for p in the simulation. If a larger value is chosen for p, it encompasses more computational effort during the covariance normal equation solution. In both of these cases, the algorithm for h and p modification is simplified as the values of either h or p need not be calculated. Upon comparing the error obtained in Figs. 34(b) and 34(c) with the error profile in Fig. 33(b) for the unmodified experiment, it can be seen that the latter is less than the former two. This fact demonstrates that the adaptation of the parameters h and p has indeed reduced the error in the prediction.

In summary, the presented tool TEMB is an efficient method to calculate the end-to-end available bandwidth between two network points, either on a given path or on the min-hop path between the two points. In addition, TEMB is very accurate, reliable, scalable and non-intrusive.

Chapter 7

Inter-domain Management

For effective end-to-end QoS guarantees, TEAM's management capabilities should be extended for inter-domain operation. In this chapter, a new scheme for estimating the traffic on an inter-domain link and forecasting its capacity requirement, based on a measurement of the current usage, is proposed. The method allows an efficient resource utilization while keeping the number of reservation modifications to low values. The scheme for resource allocation is split into two steps. In the first step, a noisy measure of the aggregate traffic is used to evaluate the number of flows and the second step is based on the forecast of the evolution of the traffic requests. This scheme was introduced in [112]. An improvement to the first step was provided in [113].

This chapter is organized as follows: The motivation for the development of the new resource allocation scheme is given in Section 7.1. In Section 7.2, related work for the proposed resource allocation scheme is presented. Then, in Section 7.3, the formulation of the traffic estimation and resource allocation scheme is presented. Performance evaluation of the proposed scheme is presented in Section 7.4.

7.1 Motivation

TEAM is responsible for allocating preferred service to users as requested, and for configuring the network routers with the correct forwarding behavior for the defined service for each class. It helps in dynamic resource management for the DiffServ classes. To be a comprehensive manager for provisioning of end-to-end QoS guarantees, TEAM's functionalities should be extended for inter-domain operation. Inter-domain tasks cover the specification of bilateral Service Level Agreements (SLAs) with neighboring domains and managing the boundary routers to police/shape the incoming/outgoing traffic to adhere to the SLAs. The TEAM of a transit domain has to reserve resources between the ingress and egress points of the domain. End-to-end QoS can then be achieved by concatenation of the intra- and inter- domain reservations. Several protocols have been suggested for inter-domain resource management signaling, such as RSVP [3], SIBBS [114] etc.

When an allocation is desired for a particular flow, a request is sent to the TEAM of the concerned AS. The request specifies the service type, target rate, maximum burst, and the time period when service is required. In general, the request can be originating from an end-user or a neighboring region's TEAM. If the request is valid, the TEAM finds the route along which request will be forwarded and then verifies the existence of sufficient unallocated bandwidth on the link with the next AS to satisfy the requested QoS. If the request passes these tests, the network resources are correspondingly provisioned. In the case of a transit AS, the TEAM has to verify sufficient resources within the network and on the downstream link. Under the DiffServ architecture, user flows are aggregated on the boundary nodes. Consequently, resource allocations are made on an aggregate basis in the core. Provisioning on the Edge Routers (ERs) can be easily determined based on the SLS in place with the customer devices. To guarantee the end-to-end QoS requirements of a request, TEAM makes bilateral agreements with its neighboring TEAMs, rather than multilateral agreements with all possible destination domains. An important requirement of the inter-domain agreements is that the changes involved should be less frequent and should be on a time-scale larger than the individual flow variations. If not satisfied, the scalability of the provisioning scheme is compensated.

Current resource allocation methods can be either *off-line* or *on-line*. Off-line, or static, methods determine the allocation amount before the transmission begins. These approaches (*e.g.* [61]) are simple and predictable but lead to resource wastage. On-line, or dynamic, methods (*e.g.* [62, 63, 64]) periodically renegotiate resource allocation based on predicted traffic behavior. These methods undergo a large number of re-negotiations.

7.2 Related Work

Conventional approaches for resource allocation rely on pre-determined traffic characteristics. Network traffic can be divided into elastic (*e.g.,* TCP) and non-elastic streaming (*e.g.,* UDP) traffic [115]. These two types differ in their requirements from the network. Packet level characteristics of elastic traffic are controlled by the transport protocol and its interactions with the network, whereas the non-elastic flows have inherent rate characteristics that must be preserved in the network to avoid losses. The source characteristics may not be known ahead of time, specified parameters may not characterize the source adequately or a large number of parameters may be required to define traffic characteristics, thus making the pre-determined traffic characteristic-based resource allocation inefficient.

One scheme for resource provisioning is to have a bandwidth "cushion", wherein extra bandwidth is reserved over the current usage. As proposed in [65], if the traffic volume on a link exceeds a certain percentage of the agreement level, it leads to a multiplicative increase in the agreement. A similar strategy is proposed in case the traffic load falls below a considerable fraction of the reservation. This scheme satisfies the scalability requirement but leads to an inefficient resource usage. This drawback can become increasingly significant once the bandwidth requirements of the users become considerable.

In this chapter, an on-line scheme called Estimation and Prediction Algorithm for TEAM (EPAT) to forecast the bandwidth utilization of inter-domain links is proposed. The scheme is designed to be simple, yet effective, when compared to more advanced prediction algorithms because it is intended to be used by TEAM and so the design goal is simplicity. The first step of the scheme is to perform an optimal estimate of the amount of traffic utilizing an inter-domain link based on a measurement of the instantaneous traffic load. This estimate is then used to forecast the traffic bandwidth requests so that resources can be provisioned between the two domains to satisfy the QoS of the requests. The estimation is performed by the use of Kalman Filter [66] theory while the forecast procedure is based on deriving the transient probabilities of the possible system states. As shown later, this scheme outperforms the current resource reservation mechanism ("cushion-based" allocation [65, 67]) employed by network operators and also some other prediction schemes based on Gaussian [68, 69] as well as local maximum [70] predictor. The proposed scheme reduces bandwidth wastage without introducing per-flow modifications in the resource reservation. Kalman Filters have been previously applied to flow control in high-speed networks. In [48], Kalman Filter was given for state estimation in a packet-pair flow control mechanism. In [116], Kalman Filter was used to predict traffic in a collection of VC sources in one VP of an ATM network. This work distinguishes itself from previous work as the Kalman Filter is used as an optimal estimation algorithm, instead of filtering or smoothing and it is an input to the capacity forecast step.

7.3 Traffic Estimation and Resource Forecast

The goal is to estimate the level of traffic between two network domains, for a given traffic class, based on a periodic measurement of the aggregate traffic on the inter-domain link. the traffic measurements are performed at discrete time-points mT, $m = 1, 2, \ldots, M$ for a given value of T. The value of T is a measure of the granularity of the estimation process and denotes the renegotiation instants. Larger values imply less frequent estimation which can result in larger estimation errors. In EPAT, periodic renegotiation instants have been assumed. At the time instant m (corresponding to mT), the aggregate traffic on the inter-domain link $l(1, 2)$ for a given traffic class in the direction $AS1$ to $AS2$ is denoted by $y(m)$. For the duration $(0, MT]$, the number of established sessions that use $l(1, 2)$ is N. For each session, flows are defined as the active periods. So, each session has a sequence of flows separated by periods of inactivity. For a given traffic class, $x(m)$ is the number of flows at the instant m and by $x(mT + t), t \in (0, T]$ the number of flows in the time interval $(mT, (m + 1)T]$, without notational conflict. Clearly, $x(m) \leq N$ and is not known/measurable. Each flow within the traffic class has a constant rate of b bits per second. So, nominally, for a traffic class

$$y(m) = bx(m). \tag{39}$$

The emphasis here is on a single traffic class and its associated resource utilization forecasting. To consider a scenario with different classes of traffic with different bandwidth requirements, the same analysis can be extended and applied for each class. Here, the resource allocation is provided for the DiffServ Expedited Forwarding (EF) classes, for which the assumption of constant resource requirement is valid. The underlying model for the flows is assumed to be Poisson with exponentially distributed inter-arrival times (parameter λ) and durations (parameter μ). Characteristics of IP traffic at packet level are notoriously complex (self-similar). However, this complexity derives from much simpler flow level characteristics. When the user population is large, and each user contributes a small portion of the overall traffic, independence naturally leads to a Poisson arrival process for flows [117, 115]. The following analysis has been carried out using this assumption and the experimental results show that the capacity forecast is very close to the actual traffic.

The EPAT scheme for resource allocation is split into two steps. In the first step, a rough measure of the aggregate traffic $y(m)$ is taken and it is used to evaluate the number of flows through the Kalman Filter estimation process. In the second step, the resources $R(m)$ to be allocated on the link $l(1,2)$ for the time $t \in (mT, (m+1)T]$ are determined based on the forecast of the evolution of $x(m)$. First, a brief introduction to the Discrete Kalman Filter theory is given.

7.3.1 Discrete Kalman Filter

The Kalman filter [66] is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. It implements a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance - when some presumed conditions are met. It estimates a process by using feedback control.

It supports estimation of past, present and future states, even if the knowledge of the precise nature of the modeled system is lacking. The Kalman filter tries to estimate the state x of a discrete-time controlled process. The system is described by the state vector x that is observed at times $m = 0, 1, \ldots$ and is governed by the linear difference equation

$$x(m+1) = Ax(m) + Bu(m) + w(m). \tag{40}$$

Each of the observations is, however, corrupted by noise and thus, the actual measurement $\bar{y}$ at times $m = 0, 1, \ldots$ is given by

$$\bar{y}(m) = Cx(m) + v(m) \tag{41}$$

The random variables $w(m)$ and $v(m)$ represent the process and measurement noise, respectively. They are independent zero-mean Gaussian white noise processes [118].

$$\begin{aligned}
E[w(m)w(k)] &= \begin{cases} \sigma_w^2 & \text{k=m} \\ 0 & \text{otherwise} \end{cases}, \\
E[v(m)v(k)] &= \begin{cases} \sigma_v^2 & \text{k=m} \\ 0 & \text{otherwise} \end{cases}, \\
E[v(m)w(k)] &= 0.
\end{aligned}$$

The parameter A in Eq. 40 relates the states at previous and current time-steps, in the absence of either a driving function or process noise. A is assumed to stay constant over the analysis. The parameter B relates the optional control input u to state x whereas the parameter C in Eq. 41 relates the state to the measured value. The objective of the Kalman filter is to obtain a "best" estimate, in some suitable sense, of $x(m)$ based on the observed values $\overline{y}(m)$ and knowledge of the statistical properties of the process and measurement noise. According to the Kalman filter mechanism, the best estimate for $x(m)$ can be obtained recursively from the previous best estimate and its covariance matrix. Thus, the Kalman filter can be divided into two steps: *prediction* and *correction*. The first step is responsible for projecting forward in time the current state to obtain *a priori* estimates of the states and

```
State Equation : $\boldsymbol{x(m) = A(m-1)x(m-1) + Bu(m-1) + w(m-1)}$
Observation Equation : $\boldsymbol{y(m) = C(m)x(m) + v(m)}$
Step 1: Initialization
    $\hat{x}(0|0) = E[x(0)]$
    $P(0|0) = E[x(0)x(0)^T] = \sigma^2_{x(0)}$
Step 2: Computation
    for m = 1,2, ...
    prediction step:
        $\hat{x}(m|m-1) = A(m-1)\hat{x}(m-1|m-1) + Bu(m)$
        $P(m|m-1) = A(m-1)P(m-1|m-1)A^T(m-1) + \sigma^2_w$
    correction step:
        $k(m) = P(m|m-1)C^T(m)[C(m)P(m|m-1)C^T(m) + \sigma^2_v]^{-1}$
        $\hat{x}(m|m) = \hat{x}(m|m-1) + k(m)[y(m) - C(m)\hat{x}(m|m-1)]$
        $P(m|m) = \{I - k(m)C(m)\}P(m|m-1)$
```

Figure 35: Discrete Kalman Filter.

covariance in the next time step. The second step is responsible for the feedback to obtain an improved *a posteriori* estimate. The Kalman filter is summarized in Figure 35.

7.3.2 Traffic Estimation

The only measurable variable in the system is $\bar{y}(m)$ which is a measure, corrupted by noise, of the aggregate traffic on the link. Nominally, $x(m) = y(m)/b$, but there is no access to the correct measurements of $y(m)$, even though b is a known quantity for a particular traffic class. Thus, the Kalman filter setup will be used to evaluate $\hat{x}(m)$, an estimate of the actual $x(m)$, using $\bar{y}(m)$ the noisy measurements. Real measurement noise makes $\bar{y}(m)$ noisy. In other words, the measurements obtained from the network for the instantaneous traffic on the link can be noisy due to miscalculation, misalignment of timings etc. To use the Kalman filter setup, a relation between $x(m)$ and $y(m)$, and $x(m)$ and $x(m+1)$ is needed. To this purpose, $p_k(t), t \in (mT, (m+1)T]$ is defined to be the probability that the number of active flows at time t is k *i.e.* for $t \in (mT, (m+1)T]$

$$p_k(t) \triangleq \text{prob}\{x(t) = k\}. \tag{42}$$

The state-transition-rate diagram is shown in Figure 36. The diagram depicts transitions

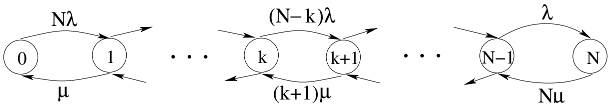


Figure 36: State-transition-rate Diagram.

among the states. From the diagram and by using queuing theory [119], the following differential equations (43-45) for the probabilities $p_k(t)$ can be obtained:

$$\begin{aligned}
\frac{dp_0(t)}{dt} &= \mu p_1(t) - N\lambda p_0(t), && (43)\\
\frac{dp_k(t)}{dt} &= (N-k+1)\lambda p_{k-1}(t) + (k+1)\mu p_{k+1}(t) \\
&\quad -(k\mu + (N-k)\lambda)p_k(t) \quad 1 \leq k < N, && (44)\\
\frac{dp_N(t)}{dt} &= \lambda p_{N-1}(t) - N\mu p_N(t). && (45)
\end{aligned}$$

The generating function $G(z,t)$ is defined as the z-transform of the probability distribution function. It aids in the computation of the mean and variance of the probability distribution. Next, the quantity $\partial G(z,t)/\partial t$ can be calculated using the Eqs. (43)-(45) as

$$G(z,t) \triangleq \sum_{j=0}^{N} p_j(t) z^j$$

$$\frac{\partial G(z,t)}{\partial t} = G(z,t)N\lambda(z-1) - \frac{\partial G(z,t)}{\partial z}(z-1)(\lambda z + \mu).$$

Utilizing the initial condition $G(z, mT) = z^{x(m)}$, *i.e.*, the number of active flows at time mT is $x(m)$, the following solution for $G(z,t)$ is obtained:

$$G(z,t) = C(z,t)^{x(m)} \left(\frac{\lambda z + \mu}{\lambda C(z,t) + \mu}\right)^N, \ \ \text{fort} \in (mT, (m+1)T]$$

where

$$C(z,t) = \frac{\lambda z + \mu - \mu(z-1)e^{-t(\lambda+\mu)}}{\lambda z + \mu - \lambda(z-1)e^{-t(\lambda+\mu)}}.$$

By the definition of the generating function and the special properties of the z-transform,

$$\begin{aligned}
E[x(m+1)|x(m)] &= \left.\frac{\partial G(z,T)}{\partial z}\right|_{z=1} \\
&= x(m)e^{-T(\lambda+\mu)} + \frac{N\lambda}{\lambda+\mu}\left[1 - e^{-T(\lambda+\mu)\}}\right]. && (46)
\end{aligned}$$

Comparing the equation with Kalman filter (Eqs. (40) and (41)) formulation

$$\begin{aligned} A &= e^{-T(\lambda+\mu)}, \\ u(m) &= 1, \\ B &= \frac{N\lambda}{\lambda+\mu}[1-e^{-T(\lambda+\mu)}], \\ C &= b. \end{aligned} \tag{47}$$

Thus, from the Kalman filter setup,

$$\hat{x}(m) = A\hat{x}(m-1) + B + k(m)[\bar{y}(m) - CA\hat{x}(m-1) - CB] \tag{48}$$

where $k(m)$ is Kalman Filter gain as defined in Figure 35. This gives an estimate of the traffic on the link currently. This estimate will be used to forecast the traffic for the purpose of resource reservation.

7.3.3 Bandwidth Request Forecasting

The optimal estimate $\hat{x}(m)$ of the number of active flows can now be used to forecast $R(m+1)$, the resource requirement on the link $l(1,2)$ between AS1 and AS2. To this purpose, for $mT < t < (m+1)T$,

$$p_i(t) \overset{\triangle}{=} \text{prob}\{x(t) = i\}, \tag{49}$$

Also define $\underline{P}$ and $\mathbf{Q}$ as

$$\underline{P} = \begin{bmatrix} p_0(t) \\ p_1(t) \\ \vdots \\ p_N(t) \end{bmatrix} \quad \mathbf{Q} = \begin{pmatrix} -N\lambda & \mu & 0 & 0 & \\ N\lambda & -[(N-1)\lambda+\mu] & 2\mu & 0 & \cdots \\ 0 & (N-1)\lambda & -[(N-2)\lambda+2\mu] & 3\mu & \\ & \vdots & & & \ddots \end{pmatrix}$$

from Eqs. (43)-(45). It can be seen that $\dot{\underline{P}} = \mathbf{Q}\underline{P}$. It is easy to demonstrate that $\mathbf{Q}$ is similar to a real symmetric matrix and reducible to a diagonal form $\mathbf{Q} = \mathbf{Y\Gamma Y}^{-1}$ where Γ is a diagonal matrix with eigenvalues of $\mathbf{Q}$ and $\mathbf{Y}$ is the matrix of corresponding right eigenvectors y. The eigenvalues can be found to be $\gamma_k = -k(\lambda + \mu)$ for $k = 0 \ldots N$ which proves that $\mathbf{Q}$ is non-positive definite, guaranteeing the existence of a solution for $\underline{\dot{P}} = \mathbf{Q}\underline{P}$. The solution, for $t \in [mT, (m+1)T]$, is given by

$$\underline{P} = \mathbf{Y}e^{\mathbf{\Gamma}t}\underline{C}, \tag{50}$$

where $\underline{C}$ is a constant vector determined from the initial condition ($x = x(m)$ at instant mT) as

$$\underline{C} = (e^{\mathbf{\Gamma}mT})^{-1}\mathbf{Y}^{-1}\underline{P}_{mT} \tag{51}$$

where $\underline{P}_{mT}$ is a vector with all 0's except the $x(m)$th element which is 1. Also define

$$\begin{aligned} \underline{\tilde{P}} &\triangleq \frac{1}{T}\int_{mT}^{(m+1)T} \underline{P}\, dt \\ &= \frac{1}{T}\mathbf{Y}\left(\int_{mT}^{(m+1)T} e^{\mathbf{\Gamma}t}\, dt\right)\underline{C} \\ &= \frac{1}{T}\begin{bmatrix} \tilde{p}_0 & \tilde{p}_1 & \cdots & \tilde{p}_N \end{bmatrix}^T \end{aligned} \tag{52}$$

using the notation that integral of a matrix is the integral of each element of the matrix. The elements $\tilde{p}_i$ of the vector $\underline{\tilde{P}}$ denote the probabilities of transitioning to state i at instant $(m+1)T$. Now define $\tilde{x}(m)$ as

$$\tilde{x}(m) \triangleq \min_{x \in [\hat{x}(m), N]} x \quad s.t. \quad \tilde{p}_x < \beta. \tag{53}$$

In words, the minimum $\tilde{x}$ greater than or equal to $\hat{x}(m)$ is found such that the probability to be in state $\tilde{x}$ during the interval $(mT, (m+1)T]$ is less than a given threshold β, in effect choosing a state greater than the current utilization estimate such that the transition probability to the state is low. Then the resource requirement is forecasted to be

$$R(m+1) = b\tilde{x}(m). \tag{54}$$

7.4 Performance Evaluation

The purpose of the simulations is to verify the accuracy of the proposed mechanisms for traffic estimation and bandwidth request forecast and compare the resource requirement

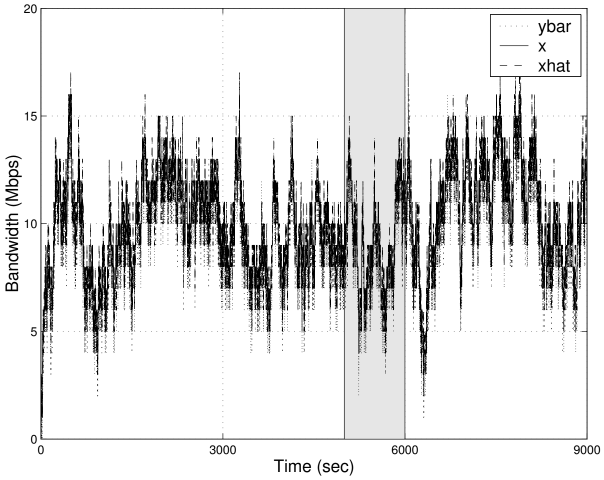
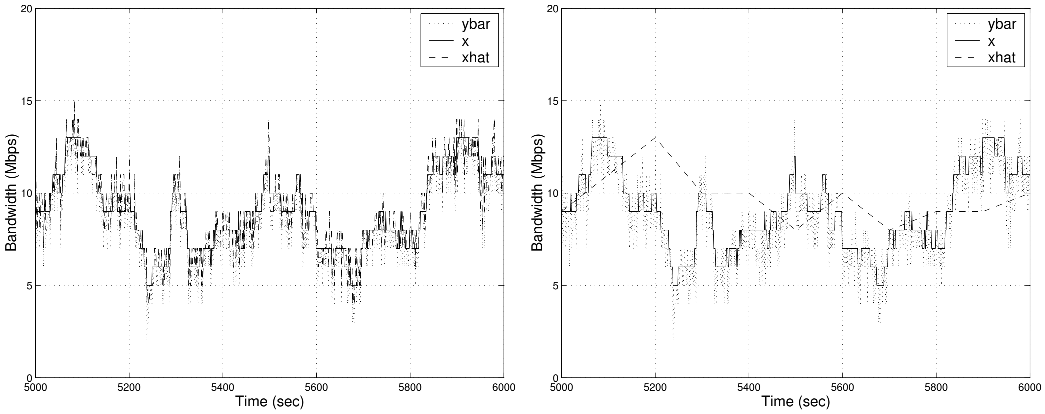


Figure 37: EPAT Estimation Performance.

forecasted using EPAT with results of other schemes. The results show how inter-domain agreements are adjusted depending on the traffic load and how closely they follow the load.

7.4.1 Estimation Performance

In the simulation, the number of established sessions, N, is assumed to be 20 with λ and μ (parameters of exponential distributions for inter-arrival time and durations of flows, respectively) of 0.005 and 0.005. In other words, for each established session, the average inter-arrival time between flows and their average duration are 200 seconds each. Shown in Figure 37 is the estimate $\hat{x}$ of the number of flows x, for a typical simulation run. The estimate is derived using the Kalman filter setup given in Eq. 48, from the noisy measurement $\overline{y}$. For this simulation, the measurement interval T was set at 1. The value of b, the constant rate for each flow of a traffic class, was set to 1 Mbps. The values for $E[x(0)]$ and $P(0|0) = E[x(0)x(0)^T]$ in the initialization step of the Kalman filter were chosen to be 1 unit. The initial choice of $E[x(0)x(0)^T]$ is not critical as long as it is non-zero because the filter will converge in any case. In the computation step, the previous values are used to compute the next values. Appropriate values were selected for process and measurement noise standard deviations as 1 and 1.5 units, respectively. If different initial seeds were chosen for $E[x(0)]$ and $P(0|0)$, the convergence time of the Kalman filter would vary, but



(a) T=1 sec

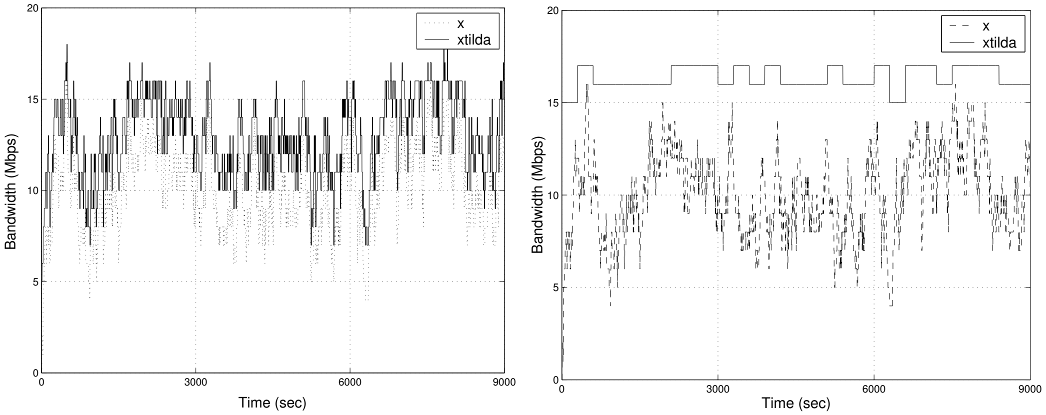
(b) T=100 sec

Figure 38: EPAT Estimation Performance (Enlarged).

the performance obtained would be similar, once the convergence is obtained. On the other hand, the variations in measurement noise standard deviation reflect in the performance of EPAT. If the standard deviation is higher, the filter is "slower" to believe the measurements and so is sluggish. If the standard deviation is smaller, the filter is "quicker" to believe the noisy measurements and follows the measurements more closely. Due to the small granularity of the estimation process, the estimated sequence has a very jagged profile in Figure 37. In Figure 38(a), the highlighted part of Figure 37 has been enlarged to show that the estimated sequence is very close to the actual traffic, despite the noise in the measured sequence. If the granularity of the estimation procedure is increased, the estimation sequence becomes more smooth (as seen in Figure 38(b)) but worse at the estimation, *i.e.*, introduces estimation errors.

7.4.2 Resource Allocation Performance

Using the estimated sequence with high granularity (Figure 37), resource requirement is forecasted using the formulation given in Section 7.3. The forecast procedure computes the probabilities $p_i(t)$ of transitioning to all possible states from the current state (Eqs. 50 and 51) and chooses the state whose transition probability is less than a threshold β as



(a) T=10 sec

(b) T=300 sec

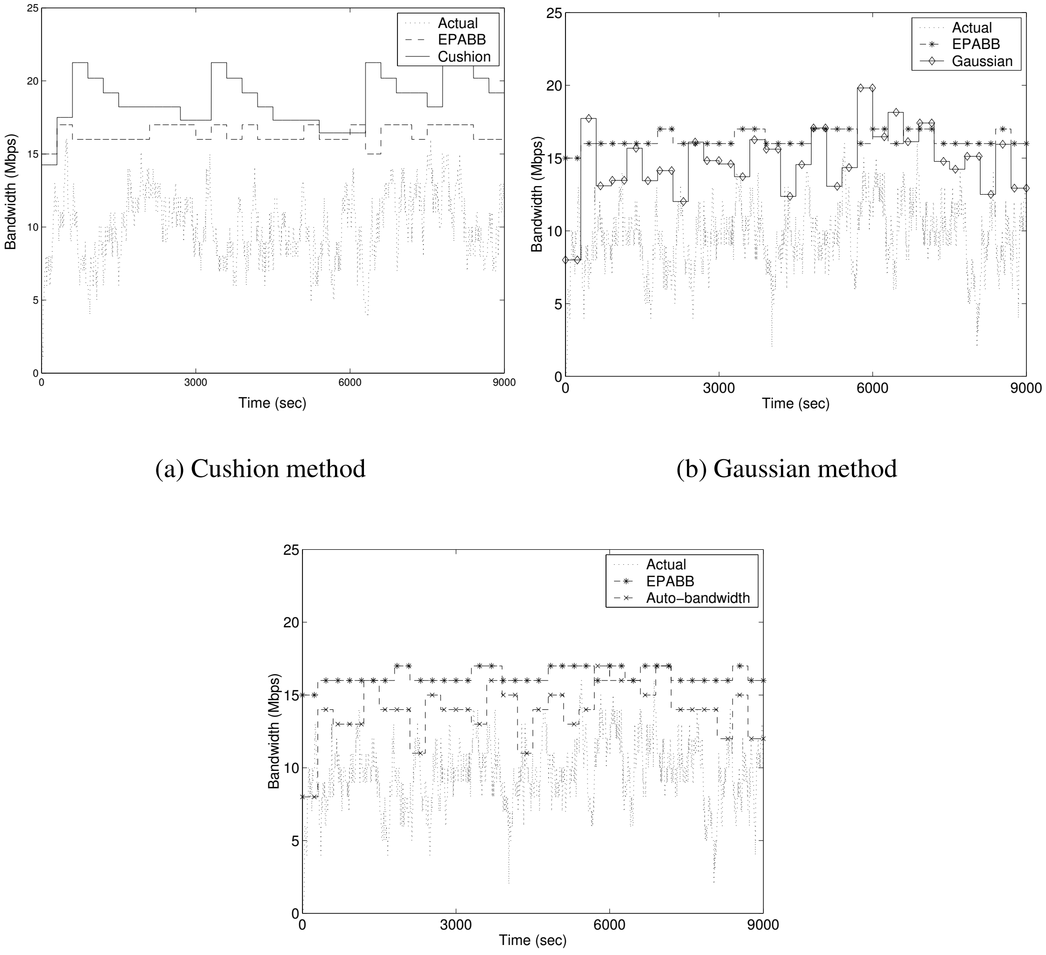
Figure 39: Forecast Performance.

shown in Eq. 53. In the simulations, β was fixed at 1%, i.e., the forecasted state is chosen such that the system has less than 1% chance of exceeding that state. Using the memoryless property of the queuing model (Poisson arrivals and exponential durations), it is easy to conclude that the probabilities $p_i(t), t \in (mT, (m+1)T])$, as defined in Eq. 49 are independent of m. This simplification helps the simulation by reducing the computation effort and time. The calculation of $\underline{C}$ and $\underline{\tilde{P}}$ (in Eqs. 51 and 52, respectively) can be performed off-line for all possible initial states and the results stored. Then the forecast process only involves a table-lookup to determine the next state at each instant for a current state based on Eq. 53.

If the forecast interval is small, there will be frequent changes in the forecasted value but less bandwidth wastage. On the other hand, if it is large, the forecasted sequence will be fairly stable at the expense of increased bandwidth wastage. Shown in Figures 39(a) and 39(b) are the forecasted sequences ($\tilde{x}$) for small and large forecast intervals, respectively. As can be seen, the forecasted sequence follows the actual traffic more closely in the first case, at the expense of the amount of signaling effort required. In the second case, the forecasted sequence has a more stable profile (*i.e.* less signaling control is needed). The mean and standard deviation of the difference between the actual traffic and forecasted capacity requirement are 2.9 Mbps and 1.0, respectively, for small forecast interval in Figure 39(a). The corresponding values are 6.35 Mbps and 2.19, respectively, for the large forecast interval in Figure 39(b). These values reflect that the variation of the forecast error (about its mean) is small but the mean error increases for large forecast intervals, which is expected.

The prediction scheme that is currently used is very simple. It is the well-known "cushion"-based method [65] of over-provisioning where an estimate of the traffic utilizing the link is derived by measurement and then the resource requirement is forecasted to be the estimate plus a cushion to accommodate any fluctuations/measurement errors. Even though this scheme causes bandwidth wastage, it is the current method to determine resource requirements for an inter-domain link due to its simplicity. The performance of EPAT can also be compared with other well-known prediction schemes. The goal of the prediction scheme utilized by a manager should be not to derive a near-perfect prediction, but to obtain an upper bound on the resource requirement which is not too conservative. This is because the resources on the links will be provisioned based on the predicted values and if the prediction is near-perfect, it can lead to blocking of new requests or degradation of service. With this aim in mind, the Minimum Mean Square Error Linear Predictor in [62] can not be employed because it tries to predict the actual value of the measured sequence.

In the following, a comparison of the performance of Estimation and Prediction Based Algorithm for TEAM (EPAT) with three different schemes for resource prediction is presented. First is the cushion-based scheme [65, 67]. Next is the prediction based on Gaussian assumption [69, 68], from the central limit theorem, that the aggregate traffic resembles a Gaussian distribution. The last one is the Autobandwidth Allocator for MPLS from Cisco [70]. The allocator concept can be used without obtaining the tunnel reserved bandwidth but instead the link utilization estimate. The comparisons are provided in Figure 40 (a), (b) and (c). The value of the forecast interval T is kept as 300sec for all the cases. To obtain the resource reservation for the cushion-based method, the traffic was first estimated using a time window measurement process with unity sampling period. This estimate was then



(a) Cushion method

(b) Gaussian method

(c) Auto-bandwidth method

Figure 40: Comparison with Other Methods.

used to calculate the resource reservation by over-provisioning a cushion of bandwidth. Whenever the traffic estimate is close to the current reservation level(watermark of 90%), the reservation level is increased in a multiplicative manner (by 25%). When the estimated traffic falls below a certain percentage of the current reservation level (80%), and stays there consistently for sometime, the reservation is reduced in an additive manner ($\gamma = \frac{1}{8}$, $\beta = 2$, $\zeta = 4$). As can be seen from Figure 40(a), the reservations achieved using this scheme are much higher than EPAT. This scheme thus leads to over-reservation of resources.

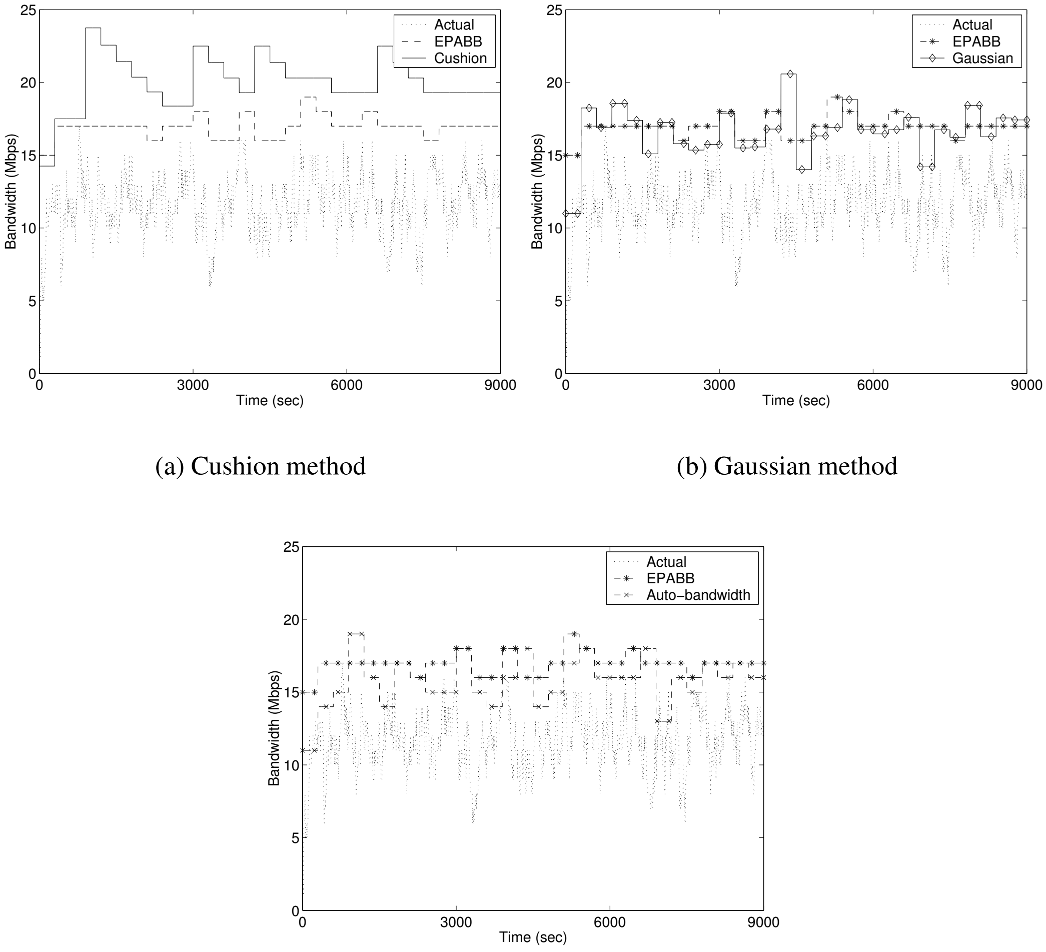
For the auto-bandwidth allocator scheme, resource reservation is obtained by first obtaining the traffic measurement using a time window with unity sampling period. The reservation for a time window is determined to be the maximum link utilization value obtained from the previous time window. For the Gaussian assumption-based allocation method, the measurements from the previous time window are obtained as before; but the allocation is determined to be the mean and 3σ from the previous window. This scheme allows for a cushion which reflects in the 3σ factor which can be reduced for a lower cushion at the expense of higher degraded QoS, as defined later. As can be seen, both these schemes infer the reservation for a time window based on the measurements from the previous time window. This introduces a lag in the reservation profile compared to the utilization. As can be seen from Figure 40(b) and (c), both schemes have low values of the over-allocation but a very high variability in the reservation profile.

Three parameters can be used to measure and compare the performance of a resource allocation scheme, namely *switching rate, bandwidth wastage* and *degraded QoS factor*. Switching rate defines the rate at which the level of the reserved resources needs to be switched to realize the desired allocation profile. It includes the increments as well as decrements in the allocation profile. Bandwidth wastage is a measure of the over-allocation of the bandwidth resources. The mean and standard deviation of the over-allocation can be used to represent its stochastic properties. Degraded QoS factor measures the percentage of bandwidth requests that will receive a degraded QoS because there is not ample reservation for them on the inter-domain link.

Comparing the reservations obtained from EPAT and cushion-based scheme for the example given in Figure 40(a), the switching rates are $1.7\mathrm{e}-3\mathrm{msec}^{-1}$ and $2.1\mathrm{e}-3\mathrm{msec}^{-1}$, respectively. The mean and standard deviation of the over-allocation by EPAT are 6.35 Mbps and 2.19, respectively whereas for cushion scheme, 8.65 Mbps and 2.84, respectively. The proposed scheme reduces the bandwidth wastage by 42% compared to the cushion-based scheme. The degraded QoS factor for both cases are 0. So, in this case EPAT reduces the over-allocation of resources without increasing the switching rate and no degraded QoS. From Figure 40(b) and (c), the auto-bandwidth allocator and the Gaussian-based allocator have much higher switching rates ($2.67\mathrm{e}-3\mathrm{msec}^{-1}$ and $3.22\mathrm{e}-3\mathrm{msec}^{-1}$, respectively) and the over-allocation is considerably lower, but the degraded QoS factor is increased (1.95% and 1.43%, respectively) due to the phase lag involved. For high QoS demanding traffic, the degraded QoS can be a problem.

7.4.3 Robustness and Sensitivity Analysis

Next, the robustness of EPAT is verified. Two tests are performed to analyze how the proposed scheme fares if the assumptions made for the analysis are removed. For the first test, the simulated input traffic is modified such that it retains its markovian properties but the rate of arrival and durations of the bandwidth requests (*i.e.,* the parameters λ and μ) are varied while the Kalman Filter estimation still uses the old values. If the Kalman Filter estimation procedure overestimates the associated parameters of the traffic, the estimation will contain error in the form of increased overestimation of resources, which is not as harmful as underestimation or increased degraded QoS factor. On the other hand, the estimation procedure should be robust to the underestimation of the parameters. In Figure 41 and Figure 42, two such cases are demonstrated. From the results given in Figure 41(a), it can be inferred that for a 40$\%$ underestimation, in the estimation procedure, of λ used for traffic generation, the switching rate of EPAT is still lower than the cushion-based scheme (compare $1.67\mathrm{e}-3\mathrm{msec}^{-1}$ and $2.11\mathrm{e}-3\mathrm{msec}^{-1}$) while the degraded QoS factor remains at 0$\%$ for both. The bandwidth wastage by EPAT is now 36% lower than the cushion-based method. The autobandwidth allocator and the Gaussian-based allocator have relatively higher switching rate and degraded QoS factor with a lower wastage of bandwidth. This test shows the tolerance of the proposed approach to misjudgment of traffic characteristics. Another scenario was also tested where both λ and μ for the traffic are underestimated in the Kalman Filter estimation. From the results given in Figure 42(a), for a 40% variation



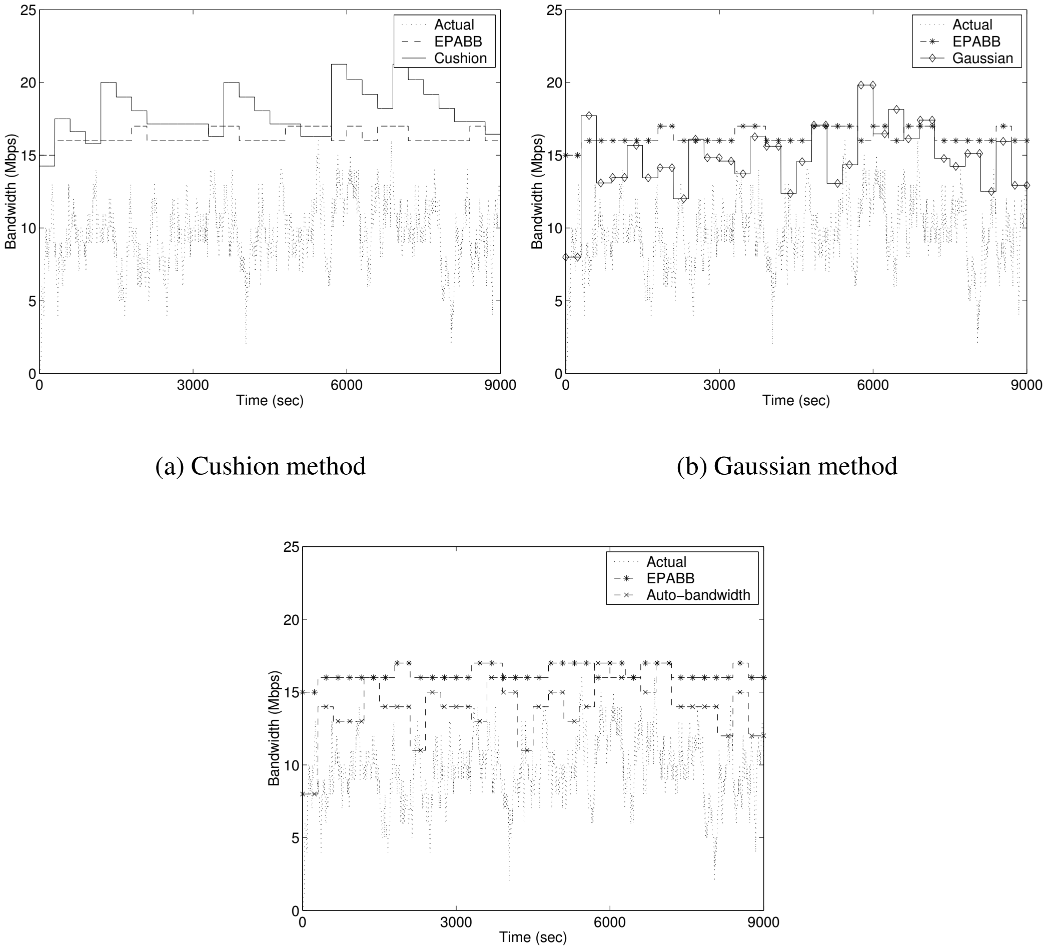
(a) Cushion method

(b) Gaussian method

(c) Autobandwidth method

Figure 41: Increased λ.

in both λ and μ, the degraded QoS factor is still 0% for both EPAT and cushion scheme but the switching rate is lower for EPAT (compare $1.44\mathrm{e}-3\mathrm{msec}^{-1}$ and $2.56-3\mathrm{msec}^{-1}$). The results for the autobandwidth allocator and the Gaussian-based allocator are similar to the previous case. Also, the sensitivity of EPAT to variations in N, the number of established sessions was checked. Theoretically, N is known to the bandwidth broker as each session is reserved after the broker provisions resources for the session. The effects on the performance of EPAT due to variations in N are shown in Figure 43. From Figure 43(a), for a 25% underestimation of N, EPAT performs fairly well, with low switching rate and marginal degraded QoS factor. On the other hand, for a 25% overestimation of N (in



(a) Cushion method

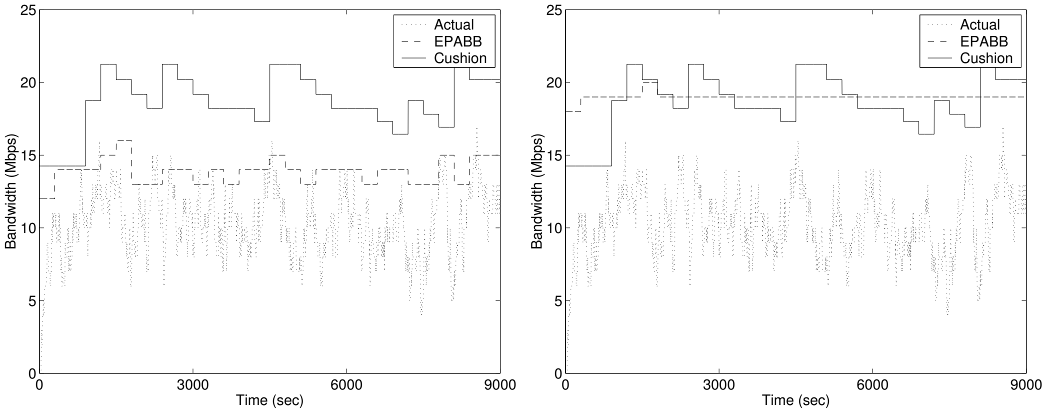
(b) Gaussian method

(c) Autobandwidth method

Figure 42: Increased λ and μ.

Figure 43(b)), the bandwidth wastage increases but it is still lower than the cushion-scheme.

The second test for verifying the robustness of the proposed scheme involves applying the scheme to an actual traffic profile obtained from measuring the traffic on a link. In this way, verification is obtained for the effect of removal of the Markovian assumption on the traffic. For this purpose, a traffic profile obtained from the publicly available traffic archives of NLANR was used. NLANR is an organization that provides technical, engineering, and traffic analysis support to high performance connections sites. This obtained profile is used as input to the Kalman Filter estimator and subsequently the capacity predictor. The values



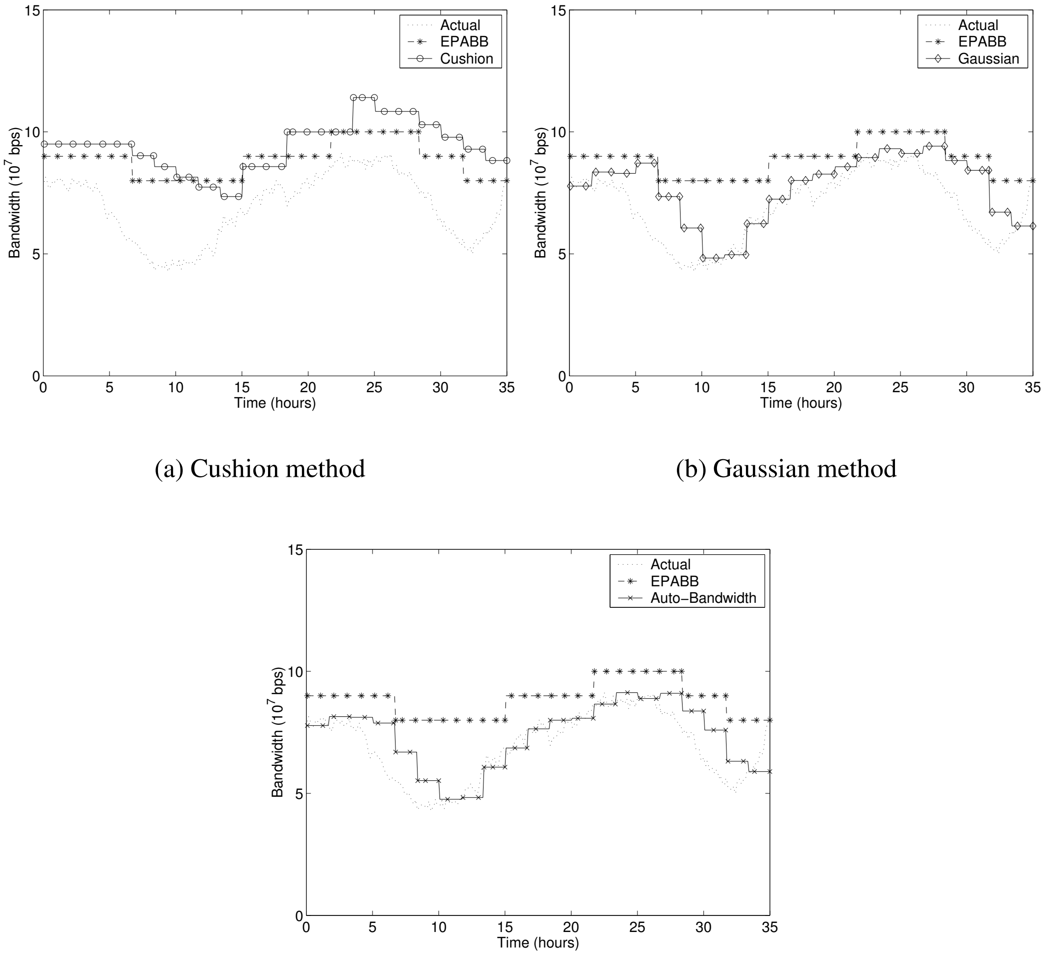
(a) N=15

(b) N=25

Figure 43: Sensitivity of EPAT to N.

for N, λ and μ can be derived from observing the traffic for some time in the past. N is assumed to be known from the SLA for the established sessions. The values of λ and μ can be derived by averaging the inter-arrival times and the inter-departure times. As the system is modeled as $M/M/N$, the average inter-arrival time is approximately $N\lambda\mu/(\lambda+\mu)$ and the average inter-departure time is $N\mu^2/(\lambda+\mu)$. The values for λ and μ need not be changed frequently as the insensitivity of the EPAT performance to the estimation error has already been shown. Figure 44(a), (b) and (c) show that the proposed scheme is able to predict the capacity requirement well and has low switching rate, bandwidth wastage and degraded QoS factor. The figures compare the performance of EPAT to the other three schemes. The traffic profile used in the figure has wide variations. Nevertheless, the EPAT scheme is able to predict the resource requirements very efficiently, without any changes in the prediction procedure or parameters.

In conclusion, the proposed scheme to estimate the traffic on an inter-domain link by the use of Kalman Filter and then forecast the capacity requirement at a future instant by the use of transient probabilities of the system states is very efficient and robust to parameter estimation errors. Switching rate, bandwidth wastage and degraded QoS factor have been used as metrics to evaluate the performance of the proposed scheme. The robustness of the



(a) Cushion method

(b) Gaussian method

(c) Auto-bandwidth method

Figure 44: NLANR Traffic Capacity Prediction.

scheme was also verified by using an actual traffic pattern.

Chapter 8

TEAM Implementation and Performance

The Traffic Engineering Automated Manager was introduced in Chapter 2. In the previous chapters, various algorithms for the efficient management of the DiffServ/MPLS domain have been presented. In this chapter, the implementation details of the TEAM software are presented. Also, the software is tested in different traffic scenarios to evaluate its efficiency. The implementation architecture was first introduced in [120].

This chapter is organized as follows: In Section 8.1, the implementation of TEAM is described and the performance of TEAM is demonstrated in Section 8.2.

8.1 TEAM Implementation

TEAM is designed for complete automated management of an Internet domain. TEAM is an adaptive manager that provides the required Quality of Service to the users, by reserving bandwidth resources, and reduces the congestion in the network by distributing the load efficiently. These goals are achieved by online measurements of the network state. TEAM is composed of a Traffic Engineering Tool (TET), which adaptively manages the bandwidth and routes in the network, a Measurement and Performance Evaluation Tool (MPET), which measures important parameters in the network and inputs them to the TET, and a Simulation Tool (ST), which may be used by TET to consolidate its decisions. These three tools work in synergy to achieve the desired network operation objectives.

A full-fledged Next Generation Internet routers physical testbed has been assembled in Broadband and Wireless Networking Laboratory (BWN-Lab) at Georgia Tech, equipped with DiffServ capable routers and switches manufactured by Cisco. The testbed comprises of a Cisco 7500 router with a Gigabit Ethernet card and a layer 3 switch Catalyst 6500

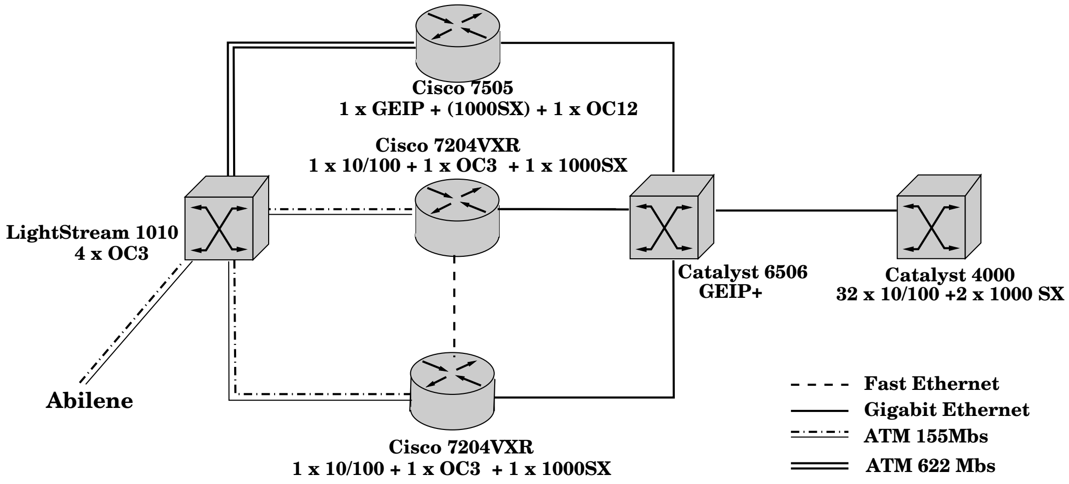


Figure 45: BWN Lab Testbed.

with an enhanced Gigabit Ethernet card and also other routers and switches. These routers and switches are widely deployed in the backbones of current high-speed networks. All the routers support MPLS and a variety of QoS technologies such as RSVP and DiffServ. Currently all devices have SNMP enabled and different measurement tools like MRTG and Netflow have been evaluated. During the analysis of MRTG, a new improved version of the tool was developed, called MRTG++, which allows managers to monitor traffic with up to 10 seconds interval, rather than the original 5 minute sampling of MRTG, providing more fine-grained detail about the state of the network. This testbed is connected via an OC3 link to Abilene, the advanced backbone network of Internet2 society. The objective of end-to-end experiments performed over this testbed is to study the advantages and disadvantages of using DiffServ in a heterogeneous traffic environment. The traffic under study is generated from voice, video and data sources. This testbed has been used as the platform to implement and test the operation of TEAM. The architecture of this testbed is shown in Figure 45.

The implementation of TEAM must be able to

- Receive a request for bandwidth reservation or LSP setup from the user.
- Implement the proposed algorithms to obtain good performance for routing, LSP setup, and preemption.

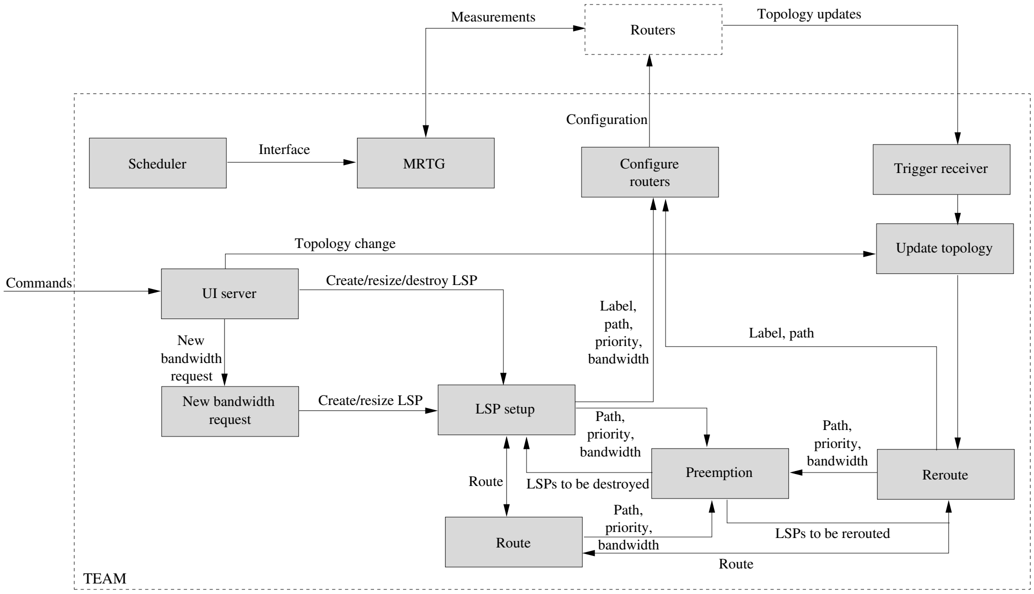


Figure 46: TEAM Top-level Design.

- Send commands and configure the testbed to create LSPs and route the traffic on the LSPs.

- Reach a decision in a timely manner to handle a large domain with 200 routers and about 20,000 LSPs.

The TEAM tool has been implemented to run on a computer with the Linux OS. It was successfully tested on RedHat 7.3, running kernel 2.4.18 on a Pentium III-800 MHz, 256MB RAM, 512MB swap space. The first version requires a TFTP server to upload the configuration to the routers. SNMP is required to ensure communication between the program and the routers. TEAM uses the net-snmp library for the communication. Version 3 of SNMP is recommended to ensure secure transmission of passwords. In order to process bandwidth measurements, RRDTool and MRTG are required. Also the GNU Scientific Library is required for matrix manipulation. The REA library [121] is used for computing k-shortest paths. The program was successfully tested on a 40 node network and 20,000 LSPs on a Pentium III computer. The top-level design of TEAM is shown in Figure 46 and the module hierarchy in Figure 47.

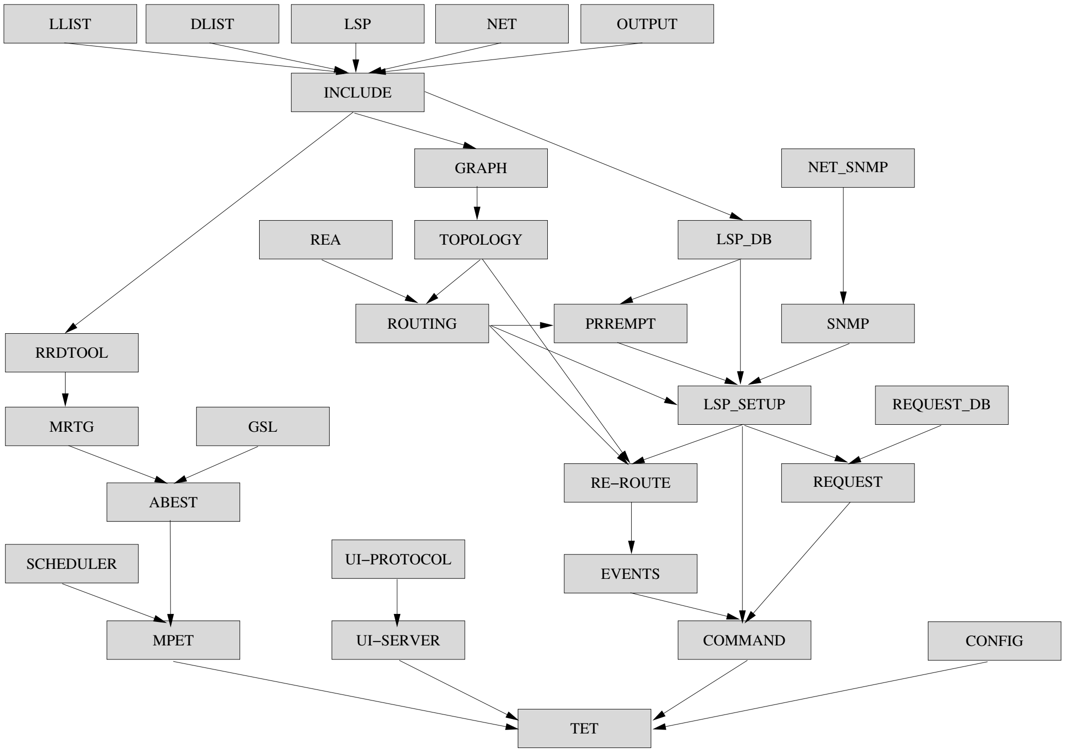


Figure 47: TEAM Module Hierarchy.

Each LSP record takes about 100 bytes in addition to the path information. It takes 20 bytes for each hop in the path. The network topology information takes about 24 bytes per node and 40 bytes per link. The LSP setup decision process takes $O(PN \log N)$ time, where P is the average path length and N is the average number of LSPs in a link.

TEAM is structured to be composed of two parts: the server and the client. The server can run at a high performance station in order to keep track of all the information of the network. The client connects to the server and sends commands using a user interface protocol. Examples of commands include the creation and destruction of LSPs, request for the topology of the network, etc.

8.1.1 Server

The server can be executed in two modes. The first one is as TET, the traffic engineering tool in which commands are received from the user and configurations are sent to the routers after a decision is made by the program. The second mode is the ST, as a simulator tool of the MPLS domain in order to study the network behavior when a decision is applied.

When the server is run in the TET mode, it stays in the background ready to receive commands from the client. It performs the following steps:

1. Load the system-wide configuration file,
2. Obtain the network topology,
3. Obtain the initial LSP topology,
4. Prompt for user's command.

At any time the program gives the option to print the current topology of the network, the LSP database and the request database. The topology shows each node and all the links that originate from it. For each link the capacity and the available bandwidth is shown. The LSP database lists all LSPs that TEAM is maintaining. For each LSP, the label, source, the interface number, destination, priority, capacity and the path is shown. Finally, the request database shows similar output. It prints all the requests that are being served by TEAM at the moment. For each request, the identification, source, destination, priority, bandwidth are shown. In addition, it also shows the label of the LSP serving the request.

TEAM can send commands to the routers using SNMP or telnet. SNMPv3 is used in order to keep communications secure. Unfortunately, current MIBs are read-only and do not allow the tool to establish LSPs directly. TEAM instructs the router to retrieve a configuration file from a TFTP server and merge it into the current setting. Although the configuration is unprotected, passwords are never sent in clear text across the network.

In the simulation mode, the server performs the same initial steps as the TET mode and then loads the command file reading one line at a time to simulate the traffic. At the end of the simulation, the tool gives the option to show the topology, the LSP database and the request database in the same way as before.

8.1.2 Client

The client is the program used to send commands to the server. It can be written and implemented in any language as long as a specific user interface protocol is used. The protocol exports the basic functionality to control the MPLS domain.

The program presents a menu with each command for a choice of user operations. For example, in order to create an LSP, the program asks which node the LSP is being originated from and the destination, priority and bandwidth. If the path is already defined, just type in each hop of the path. In order to facilitate the selection of the path, the client shows valid choices for each hop in the path. When the LSP is created by the server, the client is notified.

Similar behavior occurs for the establishment of a request. The client asks for the source, destination, priority and bandwidth and TEAM will create the request. The client can also display the topology of the network.

8.1.3 Input Files

TEAM loads some information from a set of different files. All these files are located in the input directory and their location can be modified in the configuration file. All fields are separated by a space.

The topology file contains the initial topology of the network. The initial LSP file contains the list of LSPs that will be in the database in the beginning of the execution of the program. The command file is used to send commands to the system. Before each entry, a sequence number (or time component) should be included. This number is used to identify events in the output files. It is the time component of the simulation. The configuration file controls how the program should behave. It contains ON—OFF switches for each feature of the system.

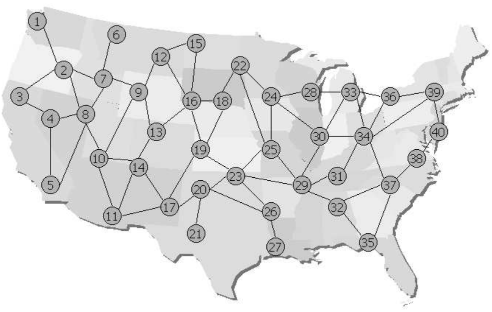


Figure 48: Network Topology.

8.2 *Performance Evaluation*

In this section, the performance of TEAM and its operation are demonstrated. The comparison of each TEAM functionality with current state-of-the-art equivalent techniques has been performed and can be found in the previous chapters. Since there are no other comprehensive network managers such as TEAM with such a diverse set of functionalities, TEAM is compared with the traditional Internet managers.

The following experimental results are obtained by simulating a network consisting of 40 nodes and 64 links, each with capacity of 600 Mbps (OC-48). This network topology is shown in Figure 48 and is based on the backbone topology of a well-known Internet Service Provider. The traffic in the network consists of aggregated bandwidth requests between node pairs having two possible priorities. The priority level 0 is the lower priority which can be preempted by the higher priority requests of level 1. These traffic requests are modeled with Poisson process arrivals and exponential durations. The simulations are divided into two broad traffic scenarios to represent significant conditions. These scenarios are characterized by different traffic loads in the network. Generalized medium and focused high traffic loads are considered to bring out the contrast in traffic conditions and observe the effects on the network performance and the different actions taken by TEAM. The generalized medium traffic load has the traffic matrix with equal values as the elements. On the other hand, the focused high load scenario is represented by a matrix where elements

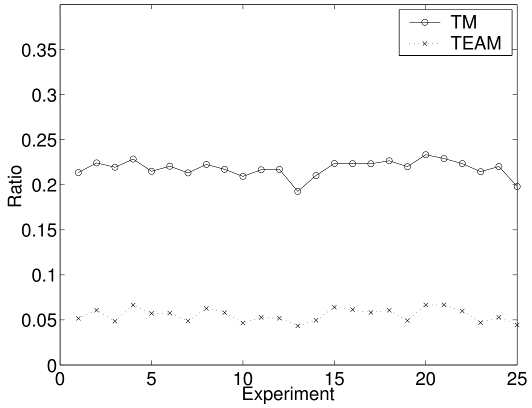
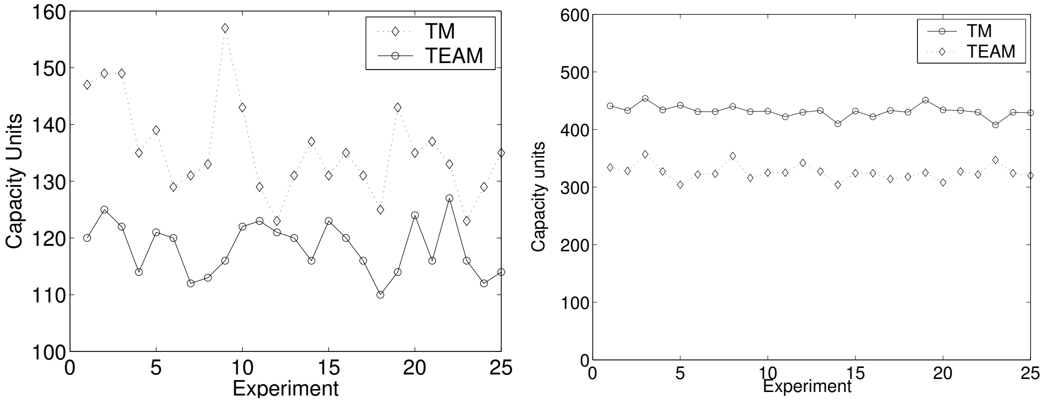


Figure 49: Rejection Ratio.

corresponding to node pairs on the opposite extremes of the network have twice the value as other node pairs.

The routing algorithm employed by TEAM, SPeCRA [86], uses many well-known algorithms for the performance comparison. This set can be modified depending on the network requirements. In the following experiments, shortest path, widest path and maximum utility based routing algorithms are used.

To evaluate the performance of TEAM as the network manager, both the network performance and the complexity associated with TEAM are analyzed. In particular, for the performance, the rejection of requests, the load distribution, the cost of network measurements, and the cost of providing the service to the requests are considered. The complexity is measured by the number of actions performed by TEAM, and by the level of the cascading effect of these actions. These metrics are compared for a network which is managed by TEAM and a network which is managed by a traditional manager (TM), like in the current Internet. In the traditional network management, the MPLS network topology is static and is the same as the physical network. In this case, the shortest path routing algorithm is used for LSP establishment, there is no LSP preemption and there are no on-line network measurements for adaptive network management.



(a) Minimum

(b) Average

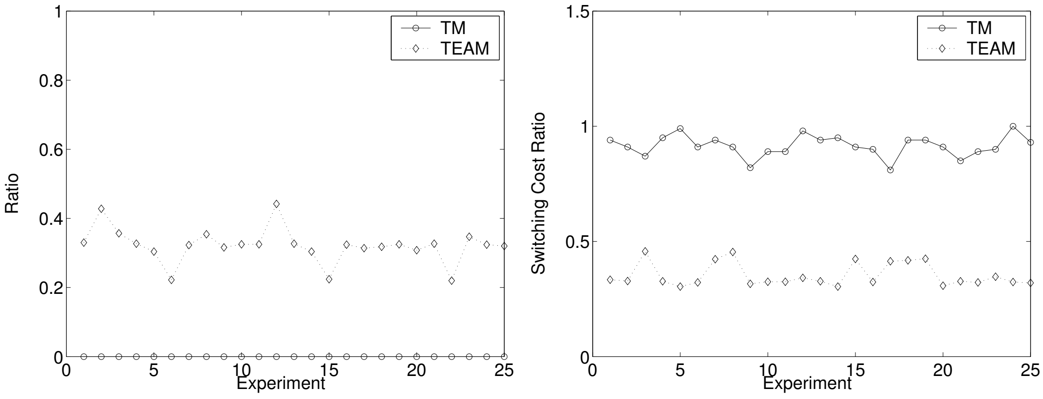
Figure 50: Available Bandwidth.

8.2.1 Generalized Medium Traffic Load

By running the experiments a few times with the generalized medium traffic load, it was observed that the LSP setup and LSP routing techniques played a major role as compared to LSP preemption. In Figure 49, the rejection ratio for the requests with and without TEAM is shown. As can be seen, the rejection is 75% lower when TEAM is managing the network. TEAM is able to achieve lower rejection due to the efficient load balancing as compared to the traditional network management.

Next, the efficiency of TEAM is demonstrated by comparing the performance with respect to the minimum and average available bandwidths for all the links in the network. In Figure 50(a), the minimum available bandwidth is shown.

In the absence of TEAM, the network links have lower minimum available bandwidth as compared to the case when TEAM is active. This is attributed to the fact that the traffic load is evenly distributed in the network using TEAM. In Figure 50(b), the average available bandwidth in the network is shown. The values for the case when TEAM is employed are higher than the case when TM is employed. This gives the false impression that the performance of TEAM in this case is worse than the TM. However, this is not correct and it is still due to the poor load balancing by the traditional network manager. In fact, when



(a) Network Measurements

(b) Service Provisioning

Figure 51: Cost.

the load is not well distributed, few links in the network are overloaded and the rejection probability becomes higher. This observation is corroborated by the high rejection ratio reported in Figure 49. Summarizing, the average available bandwidth in the network is lower using TEAM because TEAM is allowing more traffic to be carried.

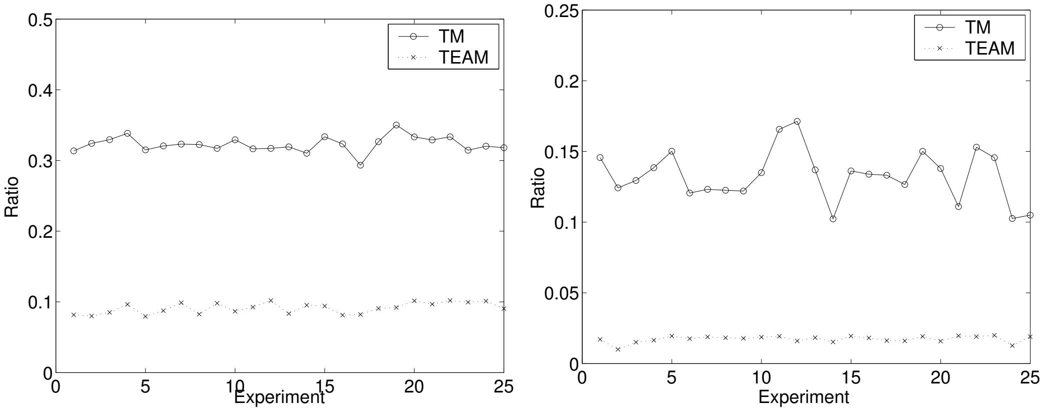
In Figure 51(a), the cost for performing network measurements is plotted. This cost is assumed to be linearly proportional to the number of available bandwidth measurements in the network. From Figure 51(a), around 30% of TEAM's actions (like LSP setup, routing and preemption) required an on-line measurement. This is compared to the TM where there is no need for network measurement since it is based on SLA contracts and nominal reservations. This measurement overhead has been limited to such low values by the filtering mechanisms in the individual TEAM techniques and it is offset by the lower rejection of the requests and consequently higher revenue. In Figure 51(b), the normalized costs of providing service in the network are plotted. The figure is mainly a representative of the traffic switching cost that can be performed in the MPLS mode or IP mode. As it is well known, it is less expensive to switch traffic in the MPLS mode as compared to the IP mode due to the simpler forwarding mechanism of the MPLS routers. The more the LSPs are created in the network, the lower is the overall switching cost for the traffic. However, the lower switching cost has to be balanced with a high signaling cost attributed to each LSP setup/re-dimension. Thus, TEAM provides an optimal number of LSPs in the network by balancing the switching and signaling costs. This optimal topology depends on the offered traffic and in this generalized medium traffic scenario, it is not as connected as the fully meshed topology. For this optimal topology, the switching cost is approximately 40% of the cost related to the static network topology. This static topology has the minimum number of LSPs as it corresponds to the physical topology.

Next, the TEAM operational load is considered. In other words, the number of actions performed by TEAM to handle the incoming bandwidth requests. 19% of the requests lead to the activation of the LSP setup/re-dimensioning procedure whereas only 0.5% of the requests were provisioned after preempting a pre-existing LSP. Most of the LSPs were routed using the shortest path routing algorithm because of the medium traffic load in the network. However, TEAM chooses other routing algorithms like widest path and maximum utility to achieve better load balancing in cases when the shortest path route is overloaded.

For this traffic load, the cascading level is always 0 as all preempted LSPs are re-established without causing any further preemptions. Thus, the cascading effects of preempting LSPs, which are undesirable, are absent in this medium traffic load scenario.

8.2.2 Focused High Traffic Load

By running the experiments with the focused high traffic load, it was observed that the LSP preemption and LSP routing techniques [86] played a major role as compared to LSP setup and capacity allocation. In the Figure 52, the rejection ratio for the requests with and without TEAM, for the two priority levels 0 and 1 are plotted. In the absence of TEAM, around 30% and 15% rejection was observed for the low and high priority requests, respectively. When TEAM is deployed in the network, the overall rejection is still 75% lower than without TEAM. However, the rejection of the high priority requests is reduced ten-fold as compared to a three-fold decrease in the low priority rejection. This considerable decrease



(a) Priority 0

(b) Priority 1

Figure 52: Rejection Ratio.

in the high priority rejection is due to the combined effect of load balancing and preemption introduced by TEAM. The rejection of low priority traffic is also reduced, but not in the same scale because only load balancing is active in this case, without preemption.

Since preemption played a significant role in this traffic scenario, the effects of various preemption policies on the network performance were observed in terms of cascading. In high traffic load scenarios, when preemption is significant, the cascading effects were minimized. When preemption was not allowed (without TEAM), the cascading effects are obviously not present. The results show that when preemption is based on priority, cascading is not critical, since the preempted LSPs will not be able to propagate preemption much further. When bandwidth is considered, fewer LSPs are preempted in each link and the wasted bandwidth is low.

Finally, the TEAM operational load is considered in this focused high traffic scenario. 35% of the requests led to the activation of the LSP setup/re-dimensioning procedure and 10% of the requests caused preemption of pre-existing LSPs. In this scenario, SPeCRA chooses the Widest Path and Maximum Utility routing algorithms more often than the Shortest Path algorithm to achieve the desired load balancing.

These results demonstrate that TEAM is an efficient manager for DiffServ/MPLS networks. The goals of a manager are QoS provisioning, efficient resource usage, and reduced risk of congestion in the network. These objectives should be achieved in variable and unpredictable traffic conditions that are characteristics of the current Internet. TEAM performs efficient resource and route management in the network for achieving the desired objectives, by using on-line measurements of network state and reacting instantly to network changes. TEAM improves network performance, at the expense of limited increases in computational and control efforts.

Chapter 9

Conclusions and Future Research Directions

In this thesis, new techniques were developed to support end-to-end Quality of Service (QoS) in DiffServ-based MPLS networks. Research contributions have been made in the following areas:

1. Automated network manager
2. LSP and setup and tear-down
3. Traffic routing
4. Link/LSP available bandwidth estimation
5. End-to-end available bandwidth measurement
6. Inter-domain management

9.1 Research Contributions

Chapter 2 presented the framework of the Traffic Engineering Automated Manager for the MPLS/DiffServ network management. The architecture of TEAM was described. Chapter 3 presented an optimal policy for LSP setup and tear-down that takes into account the bandwidth, switching and signaling costs. Whenever a new connection request arrives, a decision is made whether to setup a new LSP, to re-dimension the pre-existing LSP or to route the traffic request on a simple hop-by-hop IP route. Chapter 4 introduced a QoS traffic routing algorithm that considers multiple metrics, is scalable and operates in the presence of inaccurate information. Three algorithms were described in increasing order of complexity, in their centralized and distributed versions. The paths are chosen based on their cost which considers various metrics important for the path selection such as link available bandwidth, delay etc. Chapter 5 presented an algorithm to estimate the available bandwidth on a network link. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. Chapter 6 proposed a tool for measuring end-to-end available bandwidth over a path that can possibly span across multiple domains. The tool is efficient, easy to implement, and a combination of active and passive approaches. The tool utilizes the interface information from the MIBs in the routers along the path. Chapter 7 presented an on-line scheme to forecast the bandwidth utilization of inter-domain links, in an effort to extend the operation of TEAM for inter-domain management. The scheme is split into two steps for traffic estimation and allocation forecast. Chapter 8 presented the implementation details and performance evaluation of TEAM. Finally, an extension to the optimal policy for LSP setup and tear-down in MPLS networks for GMPLS networks is given in Appendix A. This policy decides how to route the LSP and whether a direct λSP is needed at the optical network level.

9.1.1 Automated Network Manager

An automated manager for DiffServ/MPLS networks was presented. The Traffic Engineering Automated Manager (TEAM) is comprised of a central server, the Traffic Engineering Tool (TET), that is supported by two tools: the Simulation Tool (ST) and the Measurement/Performance Evaluation Tool (MPET). The MPET provides a measure of the various parameters of the network and routers. This information is then input to the TET. Based on this measured state of the network, the TET decides the course of action, such as to vary the capacity allocated to a given LSP or to preempt a low priority LSP to accommodate a new one, or to establish the path for a traffic requiring a specified QoS. The TET automatically implements the action, configuring the routers and switches in the domain accordingly. Whenever required, the TET consolidates the decisions using the ST. The ST simulates a network with the current state of the managed network and applies the decision taken by the TET to verify the achieved performance. The TET management tasks include Bandwidth Management (LSP setup/dimensioning, LSP preemption, LSP capacity allocation) and Route Management (LSP routing). Details of the architecture and implementation were described along with performance evaluation in varying traffic scenarios.

9.1.2 Optimal Policy for LSP Setup

A new optimal decision policy that provides an online design method for MPLS networks was developed. The proposed policy is used to solve the following issue: a new request for bandwidth reservation between two routers, that are not directly connected by an LSP, arises. In this case, the decision concerning whether or not to set up a new direct LSP, modifying the current MPLS network topology, should be taken. Adding a new direct LSP requires high signaling effort, but improves the switching of packets between the two routers. The policy then decides when to setup a new LSP, when to re-dimension an existing one, or when to route the traffic on a simple hop-by-hop IP route. The optimality is derived using the Markov Decision Process formulation. For scalability reasons, a sub-optimal policy was also proposed that is easier to implement. This policy has a threshold structure and the threshold calculation takes into account the bandwidth, switching and signaling costs and depends on the network cost coefficients.

The policy was tested through simulation. Several examples were considered. Significant cases were analyzed. The results confirm that the proposed policy is effective and improves network performance by reducing the cost incurred. Simulation results also indicate that the total expected cost is similar for both the policies proving the accurateness of the sub-optimal policy. Furthermore, since a given traffic load may just be a temporary phenomenon, the policy also performs filtering in order to avoid oscillations that can be typical in a variable traffic scenario.

9.1.3 Traffic Routing

A QoS traffic routing algorithm that considers multiple metrics, is scalable and operates in the presence of inaccurate information, was presented. Numerous path choices were compared in terms of their operational costs. The cost considers all the metrics important for the path selection. The factors pertaining to the different metrics are weighed by their corresponding importance factor which can be varied from network to network. In essence, the novelty of the proposed algorithm lies in the cost structure for the LSPs and the ability to deal with the partial network state information.

The performance of the proposed algorithm was compared with the shortest path routing algorithm and found to be superior. Shortest path is the current routing algorithm used in the Internet. The proposed algorithm lowered the rejection ratio while increasing the minimum available bandwidth in the network. Thus, the algorithm achieves efficient load balancing in the network. Also, the proposed algorithm is scalable and operates under inaccurate network information.

9.1.4 Available Bandwidth Estimation

An algorithm to estimate the available bandwidth on a link was presented. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. It is a linear regression algorithm to predict the utilization of a link. The algorithm is adaptive because a varying number of past samples can be used in the regression depending on the traffic profile and it predicts the utilization and the reliability interval for the prediction.

The algorithm provides a balance between the processing load and accuracy by estimating the link available bandwidth less frequently than MRTG without a large compromise in the reliability of the estimate. The utilization estimate obtained by the algorithm provides a conservative limit on the actual utilization of the link. When the parameters used in the estimation algorithm are modified, the performance becomes worse in the sense that it does not follow the actual traffic closely but is still very conservative. Overestimation can be used as a metric to quantify the performance of the proposed scheme. The mean overestimation was observed to be 1.31 MB/s for a traffic profile with average 20 MB/s.

9.1.5 End-to-end Available Bandwidth Measurement

An accurate, scalable and flexible tool for measurement of available bandwidth along a path was presented. The tool combines the advantages of both active and passive measurement methodologies to obtain reliable measurements of the available bandwidth along a path. The functionality of the tool is distributed between both the source and destination of the path whose measurement is desired. The source sends measurement packets that collect interface information from the Management Information Bases (MIBs) in the routers along the path and are returned back by the destination to the source for analysis. The path can be pre-specified or determined hop-by-hop.

The proposed tool is very accurate, reliable, scalable and non-intrusive. Various tests were conducted to verify the efficiency of the tool. It was found that the computational effort and the intrusiveness of the scheme is highly dependent on the parameters $agree_{link}$ and $agree_{avail}$ which are application configurable and depend on the accurateness expectation of the application. If the application needs a high level of agreement before identifying the tight link, multiple attempts may be necessary to achieve the same. It was also seen that the number of attempts needed for the measurement is not very high, demonstrating that the scheme is not highly intrusive (as each attempt transmits 10 packets in 1 sec).

9.1.6 Inter-domain Management

A new scheme for estimating the traffic on an inter-domain link and forecasting its capacity requirement, based on a measurement of the current usage, was proposed. The scheme allows an efficient resource utilization while keeping the number of reservation modifications to low values. The scheme for resource allocation is split into two steps. In the first step, a noisy measure of the aggregate traffic is used to evaluate the number of flows and the second step is based on the forecast of the evolution of the traffic requests.

The performance of the proposed scheme was compared with three different schemes for resource prediction. First is the cushion-based scheme, second is the prediction based on Gaussian assumption from the central limit theorem and third is the Autobandwidth Allocator for MPLS from Cisco. The proposed scheme outperformed all three. Switching rate, bandwidth wastage and degraded QoS factor were used as metrics to evaluate the performance of the proposed scheme. Also the robustness of the scheme was verified by using an actual traffic pattern. The simulation results confirm the robustness of the scheme and show reduced wasted bandwidth.

9.2 Future Research Directions

- **Heterogeneous Large Network Management**: The automated network manager described in this thesis, TEAM, is a single domain manager. Further research would include to extend the managing area to a large network, composed of several domains. This is needed to achieve end-to-end QoS for applications. Heterogeneity and large size of networks are two main challenges facing an automated multi-domain management system. The heterogeneity may come from the coexistence of different traffic classes (e.g., best-effort and real-time), network capacity, and vendor equipments. The large network challenge results from a large number of autonomous systems in the end-to-end path.
- **MPLS and Wireless**: MPLS concepts are being applied in new domains such as wireless and voice networks. The new policies and algorithms described in this thesis were developed with a fixed wired network in mind, but can be extended to a wireless domain. New LSP setup policies would need to be developed. Other issues such as hand-off detection and decision also must be investigated.
- **Network Tomography**: TEAM is an automated manager for the DiffServ/MPLS network. Accurate network measurement techniques are needed for adaptive network management. Network tomography is a technique to infer network characteristics from end-to-end

measurements. Parameters such as link loss rates, delay statistics and topology inference are the commonly studied characteristics. However, the inference of link available bandwidths from the end-to-end measurements can be an interesting application.

Appendix A

Optimal Policy for LSP and λSP Setup

The optical network underlying the DiffServ/MPLS network also should be efficiently managed. In this chapter, a novel optimal policy is introduced to determine and adapt the Generalized MultiProtocol Label Switching (GMPLS) network topology based on the current traffic load. The Integrated Traffic Engineering (ITE) paradigm provides mechanisms for dynamic addition of physical capacity to optical networks. The objective of the proposed policy is to minimize the costs involving bandwidth, switching and signaling. The policy is derived by utilizing the Markov Decision Process theory. The new policy is split into two levels: the MPLS network level and the optical network level. In addition to the optimal policy, a sub-optimal policy and a threshold-based policy are also proposed which are less computationally intensive but have comparable performance to the optimal policy. This policy was introduced in [122].

This chapter is organized as follows: The motivation for the development of the optimal policy is given in Section A.1. In Section A.2, related work for the setup policy is presented. Then, in Section A.3, the setup problem is formulated and various definitions are explained. Next, in Section A.4, the new optimal policy is formulated and obtained. A sub-optimal policy for LSP setup is then presented in Section A.5 because the optimal policy is computationally expensive for large networks. Numerical results are analyzed in Section A.6.

A.1 Motivation

GMPLS is the proposed control plane solution for next generation optical networking. It is an extension to MPLS that enables Generalized Label Switched Paths (G-LSPs) such as

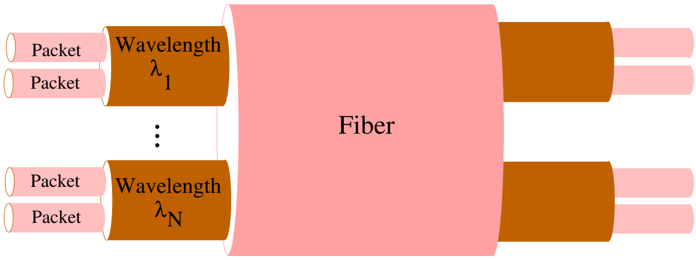


Figure 53: Link Hierarchy.

lightpaths [22], to be automatically setup and torn down by means of a signaling protocol [23]. GMPLS differs from traditional MPLS because of its added switching capabilities for lambda, fiber etc. It is the first step towards the integration of data and optical network architectures. It reduces network operational costs with easier network management and operation. The traditional MPLS is defined for packet switching networks only. It provides the advantage of Traffic Engineering (TE) when compared to other routing mechanisms, added to the improved forwarding performance. In other words, MPLS mainly focuses on the data plane as opposed to GMPLS' focus on control plane. GMPLS extends the concept of LSP setup beyond the Label Switched Routers (LSRs) to wavelength/fiber switching capable systems. Thus, GMPLS allows LSP hierarchy (one LSP inside another) at different layers in the network architecture. This concept is illustrated in Fig. 53. In this hierarchy, the packet switched link is nested inside a lambda switched link inside a fiber switched link. GMPLS also performs connection management in optical networks. It provides end-to-end service provisioning for different services belonging to different classes. Its management functionalities include connection creation, connection provisioning, connection modification, and connection deletion.

The motivation for the development of a combined method to control the topological structure of both the optical network and the MPLS networks is based on the concept of Integrated Traffic Engineering (ITE) proposed in [123]. It is a new holistic paradigm for network performance improvement, which consists of viewing the network as an integrated and cohesive system rather then a collection of independent layers. ITE attempts to tie together the key technical activities associated with network performance improvement, by taking a broad view of network performance optimization to encompass domain specific traffic routing and control, resource and capacity management, and economic considerations. The advantages of ITE include cost reduction, greater network adaptability and responsiveness to changing traffic demands, higher quality of service to end users of network services, increased efficiency of network asset utilization, and increased competitiveness. One of the objectives of ITE is to increase resource utilization, efficiency, and responsiveness by eliminating information gaps in the management of heterogeneous networks such as an IP-MPLS-over-optical network. The coordinated control and management of network resources is conducted to satisfy traffic performance requirements, improve network efficiency and reduce long term average network capital and operational costs. In particular, in the case of IP-MPLS-over-optical networks, costs can be further reduced and traffic performance enhanced by establishing direct optical connections between IP routers where substantial traffic demand exists to minimize multi-routing in the IP domain. In this way, the problem of network dimensioning, which traditionally is viewed as a long term planning problem, can be treated as a dynamical operational problem.

A.2 Related Work

Many virtual topology design algorithms [71, 72, 73] for wavelength routed optical networks have been proposed in literature. A survey of many such algorithms is given in [74]. A scheme for optical network design with lightpath protection is given in [124]. A wavelength routing and assignment algorithm for optical networks with focus on maximizing the wavelength utilization at the switches is given in [125]. However, all these algorithms design the network off-line with a given traffic matrix for the network. An on-line virtual-topology adaptation approach is suggested in [75]. This approach is concerned only with the optical network and does not relate the optical topology to the MPLS network topology.

A.3 Setup Problem Formulation

The following are defined:

- $G^F(N, L^F)$: (Physical) Topology of fibers
- $G^\lambda(N, L^\lambda)$: (Virtual) Topology of λSPs
- $G^{LSP}(N, L^{LSP})$: (Virtual) Topology of LSPs

Here, N is the set of nodes in the network and is common between the physical and virtual topologies. L^F denotes the set of links F_{ij} in the fiber network. Each $LP_{ij}^\lambda \in L^\lambda$ is a λSP between the nodes i and j (using wavelength λ), and each $LSP_{ij} \in L^{LSP}$ is an LSP between the nodes i and j. No wavelength converters are present in the network. A default λSP (LSP) is defined as the λSP (LSP) between two nodes when they are physically connected with a fiber. Thus, the default λSPs and LSPs are mapped onto the fiber network. LSP_{ij}^0 denotes the default LSP routed on the default λSP between the node pair. Only the non-default LSPs (LSP_{ij}^k) and λSPs are considered since they are candidates for redimensioning etc.

For each fiber/λSP/LSP, the following are defined:

- $C_{ij}^F/C_{ij}^\lambda/C_{ij}^{LSP}$: Capacity of fiber, λSP, LSP between nodes i and j, respectively
- $A_{ij}^F/A_{ij}^\lambda/A_{ij}^{LSP}$: Available capacity on fiber, λSP, LSP between nodes i and j, respectively

There may exist multiple λSPs between a node pair. There capacities and available capacities are distinguished by putting their wavelength specification in the superscript. B_{ij} is the total bandwidth reserved between routers i and j. Fractions of this reservation will be occupying different paths in the topology, as explained later. The following path variables are defined:

- P_{ij}^F : Minimum hop path between i and j on G^F

- P_{ij}^{λ} : Minimum hop path between i and j on G^{λ}
- P_{ij}^{LSP}: Concatenation of LSPs overlaying P_{ij}^{λ}.

The minimum hop path P_{ij}^{F} between any two nodes i and j stays constant during the analysis. This assumption is valid because addition/deletion of fibers is a part of network planning which is performed on a long-term basis. A suitable WDM technology is employed and it provides M distinct wavelengths for simultaneous use on a fiber. M is assumed constant throughout the network. The WDM technology assigns a capacity of W capacity units to each of these M wavelengths.

There exists only a single direct LSP between any node pair. This direct LSP can be routed either on a direct λSP or on the multi-λSP route. In the former case, it is denoted by LSP_{ij}^{1}, and in the latter by by LSP_{ij}^{2}. The minimum hop path on the fiber network between nodes i and j is a concatenation of the fiber links between the intermediate nodes *i.e.* $P_{ij}^{F} = \{F_{im}, \ldots, F_{nj}\}$. Similarly, the minimum hop path between nodes i and j on G^{λ} is $P_{ij}^{\lambda} = \{LP_{ih}^{\lambda i}, \ldots, LP_{kj}^{\lambda k}\}$, a concatenation of λSPs between the intermediate nodes, and $P_{ij}^{LSP} = \{LSP_{ih}^{0}, \ldots, LSP_{kj}^{0}\}$. The default LSPs are used to route MPLS traffic between two nodes when there is no direct LSP or not enough available bandwidth on the direct LSP. Thus, in an MPLS network, the bandwidth requests between i and j are routed either on a direct LSP LSP_{ij}^{k} or on P_{ij}^{LSP}, a concatenation of default LSPs overlaying P_{ij}^{λ}. P_{ij}^{λ} stays constant during the analysis. This implies that a λSP is used only to route LSPs with the same end-points as the λSP and a new LSP can not utilize a previously established non-default λSP for its routing. This assumption approximates a decentralized management architecture. The decentralized approach is widely used in the current networks and has its advantages of scalability and ease of operation. With this assumption, events in other parts of the network do not affect the local network state.

When a new bandwidth request b_{ij} arrives between routers i and j in the MPLS network, the existence of a direct LSP between i and j is checked initially. For direct LSP between i and j, the available capacity A_{ij}^{LSP} is then compared with the request b_{ij}. If $A_{ij}^{LSP} > b_{ij}$, then the requested bandwidth is allocated on that LSP and the available capacity is reduced accordingly. Otherwise, C_{ij}^{LSP} can be increased subject to bandwidth allocation constraints in order to satisfy the bandwidth request. If there exists no direct LSP between i and j, then a decision is needed whether to setup a new LSP and its according C_{ij}^{LSP}. Each time a new LSP is setup, the previously granted bandwidth allocation requests between i and j are re-routed on the new LSP. If the request can not be satisfied on the direct LSP, the request will be routed on P_{ij}^{LSP}, if there is enough available capacity on each default LSP in P_{ij}^{LSP}. If any of the default LSPs does not have the required available bandwidth, it is re-dimensioned. For this re-dimensioning, capacity is borrowed from the corresponding λSP, LP_{hk}. Since the present day λSPs are normally allocated capacities in the order of OC-192c (10Gbps), in most cases there will be enough available capacity and the bandwidth requests can be satisfied by this part of the method. However, since Internet traffic is growing exponentially and new applications are developed on a day-to-day basis, a scenario where the OC-192c capacities will be fully occupied is foreseeable. While adding new physical capacity to the traditional networks was part of the long-term network planning, with the advancement in the optical technology and the integration of the MPLS and optical control planes, the capacity addition has become a more dynamic and on-demand process. Thus, a method for setting up and tearing down λSPs depending on the bandwidth need, network performance and economic considerations is needed. This method is applied if the direct λSP capacity exceeds a threshold and a decision whether or not to setup a new direct λSP between i and j is needed.

At the time of the departure of a bandwidth request, the LSP where the request was routed is a candidate for being torn down. If the request was routed on LSP_{ij}^2, the LSP can be torn-down. However, if the request is routed on LSP_{ij}^1, the option of tearing down the LSP as well as the λSP needs to be considered. The default LSPs and default λSPs overlaying the fiber links in L^F are never torn down.

The following definitions are provided for a node pair i, j. The definitions can be extended to other node pairs independently. This assumption is valid because the events for each node pair are assumed to be independent. Thus, the subscript is dropped in the definitions henceforth.

A.3.1 Definitions

The following definitions will be used in the setup problem formulation:

- *Definition 1: Bandwidth requests*

The bandwidth requests are denoted by b. A request specifies the amount of bandwidth requested and the origin and destination end-points. Events are associated with the arrival and departures of the requests, as explained next.

- *Definition 2: Events and decision instants*

The following events are defined for the MPLS network:

- $e^{MPLS} = 0$: Arrival of a bandwidth request b
- $e^{MPLS} = 1$: Departure of request from LSP^1
- $e^{MPLS} = 2$: Departure of request from LSP^2
- $e^{MPLS} = 3$: Departure of request from P^{LSP}

and for the optical network:

- $e^{\lambda} = 0$: Arrival of LSP setup or capacity increment
- $e^{\lambda} = 1$: Departure of LSP or capacity decrement

The optical network events are generated by actions at the MPLS network. The occurrence of each event is a decision instant. The decision rules are explained later.

- *Definition 3: States*

The MPLS state vector s^{MPLS} at a given time instant for a node pair in the MPLS network is defined as

$$s^{MPLS} = [A^{LSP},\, B^{L},\, B^{P}]. \tag{55}$$

Here, B^L is the part of B that is routed on the direct LSP $LSP^k, k \in \{1, 2\}$ and B^P is the part that is routed on P^{LSP}, the concatenation of the default LSPs.

The λSP state vector s^λ at a given time instant for a node pair in the optical network is defined as

$$s^\lambda = [A^\lambda,\, B^\lambda, B^F, k], \tag{56}$$

Here, k denotes the number of λSPs between the node pair. If the capacity of the direct LSP increases beyond the λSP capacity, then another λSP is created to route the additional LSP capacity. Thus, there may exist multiple λSPs between the node pair. The first fit algorithm is used for λ assignment to the λSP. A^λ is the total available bandwidth on all the λSPs between the node pair, B^λ is the part of B that is routed on the direct λSPs between the node pair, B^F is the part of B that is routed on the λSPs in P^λ. Note that only the state of the direct λSP between the node pair is considered and not other λSPs. This is because of the assumption that the direct λSP is used only for LSPs with the same end-points.

The fiber state vector s^F at a given time instant for a node pair in the fiber network is defined as

$$s^F = [\Omega]. \tag{57}$$

where Ω denotes the set of wavelengths still available on the fiber and not being used by any λSP.

Even though the system state is split into three separate levels, the state of the system is expressed as a combination of state at all the levels. This division among the state variables has been made because the decisions (as explained later) are made independently at each level and require only the state information at that level. Note that the system state is unchanged unless an event occurs. The occurrence of an event triggers the decision policy which provides a suitable action to handle the event. Execution of the action changes the network state.

- *Definition 4: Extended states*

The MPLS and the optical state space can be extended by the coupling of the current state and the event.

$$
\begin{aligned}
S^{MPLS} &= \langle s^{MPLS}, e^{MPLS} \rangle \\
S^{\lambda} &= \langle s^{\lambda}, e^{\lambda} \rangle
\end{aligned}
$$

This extended state space $\bar{S}$ is the basis for determining the decisions.

- *Definition 5: Actions*

Assume that at time instant t, the event e occurs which has to be handled by the network. The network decides its action at both the MPLS (a^{MPLS}) and the optical (a^{λ}) levels. $a^{MPLS} = 1$ means that the direct LSP will be re-dimensioned and $a^{MPLS} = 0$ means that no action will be taken and the request is routed either on a existing direct LSP or on P^{LSP}. $a^{\lambda} = 1$ means that the λSP will be setup/torn-down and $a^{\lambda} = 0$ means that no new λSP will be setup. The combined actions at the two levels is $a = \langle a^{MPLS}, a^{\lambda} \rangle$.

- *Definition 6: Cost function*

The cost is split into two levels for the MPLS network and the optical network. The cost at each level is the sum of three components: the bandwidth cost $W_{\text{b}}(S, a)$, the switching cost $W_{\text{sw}}(S, a)$, and the signaling cost $W_{\text{sign}}(S, a)$:

$$W(S, a) = W_{\text{b}}(S, a) + W_{\text{sw}}(S, a) + W_{\text{sign}}(S, a) \tag{58}$$

with appropriate superscripts for the two levels. The cost definitions for the MPLS network are similar to the definition in Chapter 3.

At the optical level, the rate of bandwidth cost $w^{\lambda}_{\text{b}}(S, a)$ incurred depends linearly on the number of hops h^F in P^F and the capacity of the λSP.

$$w^{\lambda}_{\text{b}}(S, a) = c_{\text{cap}} k \; W \; h^F, \tag{59}$$

where c_{cap} is the bandwidth cost coefficient per capacity unit (c.u.) for the optical network. The cost is incurred for the whole capacity allocated to the k λSPs because of the large modularity in capacity allocation by the current WDM technologies.

The switching cost in the optical network depends on the number of switching operations in the optical and opto-electronic switches. The total number of switching operations is always h^F, since the physical path is fixed. The type of these operations depends on the path chosen in the optical network. Thus, the rate of switching cost is given as

$$w_{\text{sw}}^{\lambda}(S,a) = \begin{cases} E(B^{\lambda} + B^F + b) & a^{\lambda} = 1 \\ EB^{\lambda} + c_{\lambda}h^F(B^F + b) & a^{\lambda} = 0 \end{cases} \tag{60}$$

where $E = (h^F - 1)c_{\text{opt}} + c_{\lambda}$, c_{opt} is the cost coefficient for the switching of the λSP in the optical switches on the path and c_{λ} is the cost coefficient for the opto-electronic switching at the head-end of the λSP. Since the λSPs are assumed to be always formed over the physical shortest path in P^F (which stays constant), the optical switching cost coefficient is multiplied by $(h^F - 1)$, the number of successive hops with optical switching.

The signaling cost of a λSP is made up of many components. As the MPLS network, the signaling cost is incurred only when a new λSP is being created or an old one being destroyed. The components of the signaling cost include c_{sign} (the cost for signaling the information to all the relevant nodes) among others. This cost component is fixed in nature and does not depend on the network topology. The other two components of the signaling cost are proportional to h^F, the number of hops on the physical path between the nodes. They are $c_{find\lambda}$ (the cost for finding the common wavelength to be used on the fibers in P^F) and $c_{allocate}$ (the cost of allocating that wavelength to the λSP). The last component c_{moving} relates to the cost of moving the existing traffic from one λSP to another. Note that the signaling cost is instantaneous and not time-dependent. Grouping terms together:

$$W_{\text{sign}}^{\lambda}(S,a) = a^{\lambda}[c_{\text{x}} + c_{\text{y}}h^F] \tag{61}$$

A.4 Optimal Setup Policy

The optimization problem is formulated as a CTMDP as before. The expected infinite-horizon discounted total cost is

$$v_{\alpha}^{\pi}(S_0) = E_{S_0}^{\pi}\left\{\sum_{m=0}^{\infty} e^{-\alpha t_m}\left[W_{\text{sign}}(S_m,a)+\int_{t_m}^{t_{m+1}} e^{-\alpha(t-t_m)}[w_{\text{b}}(S_m,a)+w_{\text{sw}}(S_m,a)]dt\right]\right\}. \tag{62}$$

This definition can be applied to both the LSP and λSP levels. The optimal decision policy can be found by solving the optimality equations for each initial state S. The bandwidth requests arrive in the MPLS network according to a Poisson process with rate λ and the request durations are exponentially distributed with mean μ. Only some of these requests are relayed to the underlying optical network when λSP states need to be modified. Since the sampling of a Poisson process leads to another Poisson process, the λSP requests in the optical network arrive according to a Poisson process with rate λ' and are valid for exponentially distributed durations with mean μ'.

Following an approach similar to Chapter 3, the optimal LSP setup decision policy is $\pi^* = \{d^*, d^*, d^*, \ldots\}$ and the decision rule is given by

$$d^* = \begin{cases} 0 & S=\langle A,B^L,B^P,0\rangle \ A \geq b \\ a^*\langle A,B^L,B^P,0\rangle & S=\langle A,B^L,B^P,0\rangle \ A < b \\ a^*\langle A,B^L,B^P,1\rangle & S=\langle A,B^L,B^P,1\rangle \\ a^*\langle A,B^L,B^P,2\rangle & S=\langle A,B^L,B^P,2\rangle \\ 0 & S=\langle A,B^L,B^P,3\rangle \end{cases} \tag{63}$$

where

$$\begin{aligned} & a^*\langle A,B^L,B^P,0\rangle \\ & \quad = \begin{cases} 1 & c_s h^{\lambda}+c_a < v^*\left(A,B^L,B^P+2b,3\right) \\ & \qquad\qquad -v^*\left(0,B^L+B^P+b,b,3\right), \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

$$
\begin{aligned}
a^*\langle A, B^L, B^P, 1\rangle &= a^*\langle A, B^L, B^P, 2\rangle \\
&= \begin{cases} 1 & c_s h^\lambda + c_a < v^*\Big(A+b, B^L-b, B^P+b, 3\Big) \\ & \qquad\qquad -v^*\Big(0, B^L+B^P-b, b, 3\Big), \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

For the optical network, the optimality equations are:

$$
v(S) = \min_{a\in A}\left\{ r(S,a) + \frac{\lambda'+\mu'}{\lambda'+\mu'+\alpha} \sum_{j\in\overline{S}} q(j\,|S,a)v(j) \right\} \tag{64}
$$

Since the set of possible actions is finite and $r(S,a)$ is bounded, it can be proved that the optimal policy π^* is stationary and deterministic [91]. The solution of the optimality equations gives the optimal values of the expected infinite-horizon discounted total costs. The decision rule for the optimal policy at the optical level is given as

$$
d^* = \begin{cases}
a^* & S=\langle 0,0,B^F,0,0\rangle \\
0 & S=\langle 0,0,B^F,0,1\rangle \\
0 & S=\langle A^\lambda,B^\lambda,B^F,k,0\rangle \; k>0,\, A^\lambda \geq b+B^F \\
1 & S=\langle A^\lambda,B^\lambda,B^F,k,0\rangle \; k>0,\, A^\lambda < b+B^F \\
0 & S=\langle A^\lambda,B^\lambda,B^F,k,1\rangle \; k>0,\, A^\lambda < W-b-B^F \\
1 & S=\langle A^\lambda,B^\lambda,B^F,k,1\rangle \; k>0,\, A^\lambda \geq W-b-B^F
\end{cases} \tag{65}
$$

where

$$
a^* = \begin{cases}
1 & c_x + c_y h^F + \dfrac{c_{cap} W h^F}{\alpha+\lambda'+\mu'} \\
& \quad < v^*\Big(0,0,B^F+2b,0,1\Big), \\
& \quad\; -v^*\Big(W-B^F-2b, B^F+2b, 0,1,1\Big), \\
0 & \text{otherwise,}
\end{cases}
$$

The threshold structure of the optimal policy facilitates the solution of the optimality equations but still it is difficult to pre-calculate and store the solution because of the large number of possible system states. In a large network, the application of this optimal policy will no more be real-time since solving the infinite-horizon MDP problem is a time-intensive process. So, a sub-optimal policy is proposed that is easy and fast to calculate and implement in large realistic scenarios.

A.5 Sub-optimal Setup Policy

The sub-optimal policy minimizes the cost incurred between two decision instants. The sub-optimal policy for the MPLS network can be obtained as Chapter 3. For the optical network, $\pi^{\#} = \{d^{\#}, d^{\#}, d^{\#}, \ldots\}$ and the decision rule is given by

$$d^{\#} = \begin{cases} a^1 & S = \langle 0,0,B^F,0,0 \rangle \\ 0 & S = \langle 0,0,B^F,0,1 \rangle \\ 0 & S = \langle A^\lambda,B^\lambda,B^F,k,0 \rangle \; k > 0,\, A^\lambda \geq b + B^F \\ 1 & S = \langle A^\lambda,B^\lambda,B^F,k,0 \rangle \; k > 0,\, A^\lambda < b + B^F \\ 0 & S = \langle A^\lambda,B^\lambda,B^F,k,1 \rangle \; k > 0,\, A^\lambda < W - b - B^F \\ 1 & S = \langle A^\lambda,B^\lambda,B^F,k,1 \rangle \; k > 0,\, A^\lambda \geq W - b - B^F \end{cases} \tag{66}$$

where

$$a^1 = \begin{cases} 1 & B^F + b > \frac{(c_x + c_y h^F)(\alpha + \lambda' + \mu') + c_{cap} W h^F}{(h^F - 1)(c_\lambda - c_{opt})} \\ 0 & \text{otherwise,} \end{cases} \tag{67}$$

This sub-optimal policy is easy to implement in a real-time manner for a large network since it is a simple threshold-based policy where the thresholds are dependent on the network costs which are known and constant during the analysis. Thus, the thresholds can be calculated and stored a-priori and and the application of the sub-optimal policy becomes a simple comparison check for the network state. Though this sub-optimal policy is easy to implement, it is still restricted in its physical application because of the assumption that a λSP can only be used by LSPs with same end-points. An improvement to this policy can be achieved with a centralized approach where information is available about the whole network. Thus, another sub-optimal policy for λSP establishment is proposed. In this policy, the assumption is removed and intermediate λSPs are used for longer LSPs. The algorithm for this threshold-based sub-optimal policy for LSP and λSP topology adaptation is given in Figure 54. This policy achieves lower overall operational cost at the expense of increased management effort for maintaining the global network state. In this policy, B_{Th}^{MPLS} is equal to the threshold in the MPLS network and B_{Th}^{λ} is given as:

$$\begin{aligned} B_{Th}^{\lambda} = & \frac{(c_x + c_y h^F - c_x \beta - c_y \sum_{i \in \beta} h_i^F)(\alpha + \lambda' + \mu')}{(h^{\lambda} - 1)(c_{\lambda} - c_{opt}) + c_{cap} \sum_{i \in \eta} h_i^F} \\ & + \frac{c_{cap} W (h^F - \sum_{i \in \beta} h_i^F)}{(h^{\lambda} - 1)(c_{\lambda} - c_{opt}) + c_{cap} \sum_{i \in \eta} h_i^F} \end{aligned} \tag{68}$$

where β is the total number of λSPs in P^{λ} that do not have enough available bandwidth and η is the number of λSPs that do not need modification. Also, h_i^F denotes the number of fibers corresponding to the λSP i among the β λSPs to be re-dimensioned. The relations $\beta + \eta = h^{\lambda}$ and $\sum_{i \in \beta, \eta} h_i^F = h^F$ were used to derive the threshold. This threshold has been calculated by a cost comparison among the options of creating a direct λSP and re-dimensioning intermediate λSPs for LSP request.

A.6 Performance Evaluation

If the threshold-based policy is applied to a network where no additional λSPs have been added yet, and a new LSP setup request arrives, if only one out of the h λSPs needs re-dimensioning, the threshold for creation of a direct λSP becomes $\{c_y(\alpha + \lambda + \mu) + c_{cap} W\}/\{c_{\lambda} - c_{opt} + c_{cap}\}$. If two λSPs need re-dimensioning, the threshold becomes $\{c_y(\alpha + \lambda + \mu) c_{cap} W - \frac{c_x}{h-2}\}/\{\frac{h-1}{h-2}(c_{\lambda} - c_{opt}) + c_{cap}\}$. It is easy to see that the former expression is larger than the latter. Thus, it is faster to create a new direct λSP if more λSPs need re-dimensioning. This observation is very intuitive as re-dimensioning larger number of λSPs implies larger signaling cost. If all the λSPs need to be re-dimensioned to accommodate the LSP, i.e., $\beta = h^{\lambda}$, the threshold for creation of a direct λSP becomes $-c_x/(c_{\lambda} - c_{opt})$. As this value is less than zero, it implies that the direct λSP will be created even for a very small bandwidth request.

For the simulations, the physical topology of Figure 55 is used. Each node represents

```
LSP & Lightpath Setup/Re-dimensioning Policy
At time $t$, $\mathbf{s^{MPLS}} = [\mathbf{A^{LSP}}, \mathbf{B^{L}}, \mathbf{B^{P}}], \mathbf{A}^{\lambda}, \mathbf{B}^{\lambda}, \mathbf{B^{F}}, \mathbf{k}]$ and event $\mathbf{e^{MPLS}}$ occurs
Case 0: $\mathbf{e^{MPLS}}$ = Arrival of request b
    If direct $LSP^k, k \in \{1, 2\}$ exists with enough available bandwidth
        Request is accepted and routed on $LSP^k$
    Else
        If all LSPs $\in P^{LSP}$ have enough available capacity
            LSP Check: If total traffic between nodes exceeds $B_{Th}^{MPLS}$
                If direct $LSP^1$ exists
                    If $LP$ does not have enough available capacity, $LP$ is re-dimensioned
                    $LSP^1$ is re-dimensioned and request is accepted and routed on $LSP^1$
                If direct $LSP^2$ exists
                    If total traffic between nodes exceeds threshold in (67)
                        $LP$ and $LSP^1$ are created and request is accepted and routed on $LSP^1$
                If no $LSP^k$ exists, $LSP^2$ is created and request is accepted and routed on $LSP^2$
            Else request is accepted and routed on $P^{LSP}$
        Else
            Identify all default $LSP_{hk}^0 \in P^{LSP}$ without enough available capacity.
            Let $\alpha$ be the number of such LSPs
            For each such $LSP_{hk}^0$
                If the corresponding $\lambda$SP has enough available capacity $LSP_{hk}^0$ is re-dimensioned
                Else identify $\lambda$SP $LP_{hk}$
            Let $\beta$ be the total number of $\lambda$SPs without enough available capacity
            Let $\eta$ be the number of $\lambda$SPs with enough available capacity ,i.e., $\beta + \eta = h^{\lambda}$
            If total traffic on $LSP^k$ exceeds $B_{Th}^{\lambda}$
                New direct $LP$ and $LSP^1$ are created. Request accepted and routed on $LSP^1$
                Topologies $P^{\lambda}$ and $P^{LSP}$ are modified
            Else
                Create $\beta$ new $\lambda$SPs $LP_{hk}$
                $\beta$ default $LSP_{hk}^0$ are re-dimensioned
                Jump to LSP Check
Case 1: $\mathbf{e^{MPLS}}$ = Departure of b from $\mathbf{LSP^1}$
    If $B^L = b$ and $B^P = 0$ and $F$ not exists
        $LSP^1$ and $LP$ are torn-down
Case 2: $\mathbf{e^{MPLS}}$ = Departure of b from $\mathbf{LSP^2}$
    If $B^L = b$ and $B^P = 0$ and $F$ not exists
        $LSP^2$ is torn-down
Case 3: $\mathbf{e^{MPLS}}$ = Departure of b from $\mathbf{P^{LSP}}$
    Adjust $B^P$ accordingly
```

Figure 54: Threshold-Based Setup/Re-dimensioning Policy.

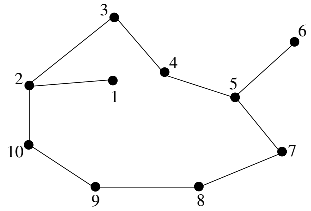


Figure 55: Network Topology.

an LSR and each edge represents a fiber link connecting two LSRs. A capacity of 10Gbps is assigned to each λSP. The cost coefficients are chosen as $c_s = c_y = 2.5, c_a = c_x = 2.5, c_b = c_{cap} = 1, c_{ip} = c_\lambda = 0.35, c_{mpls} = c_{opt} = 0.25$. The values for λ and μ are chosen such that the traffic in the network is increasing. The values of λ' and μ' are obtained by observing the statistics of the MPLS network under the optimal policy for LSP setup.

The initial topologies G^F, G^λ and G^{LSP} coincide. Homogeneously increasing amount of traffic is offered to the node pairs 2-5, 2-6, 2-7, 2-8, 10-5, 10-6, 10-7, and 10-8 and the network topology is observed over time. The LSP evolution profile shows that the longer LSPs tend to be established first. From the λSP evolution, for a given h^λ the setup threshold decreases with increasing β. The second observation is that for a given β, the threshold increases with increasing h^λ.

If the capability to add physical capacity on demand was not available, then the rejection of bandwidth requests would be very high in current networks as the traffic continues to grow. The performance of the proposed policies is compared with three well-known heuristics. In all the heuristics, a fully connected λSP network is pre-established. Thus, there is no need for λSP topology adaptation. Heuristic 1 establishes a fully connected LSP network before the network is operational. In this way, all the virtual topologies are fixed a-priori and can not be adjusted to the traffic demands. This heuristic leads to resource wastage as the reserved resources are not necessarily utilized. Heuristic 2 re-dimensions the LSPs every time there is a bandwidth request. The re-dimensioning is such that the the LSP size exactly fits the bandwidth request. This heuristic leads to large number of topology modifications. Heuristic 3 tries to reduce the number of modifications by re-dimensioning

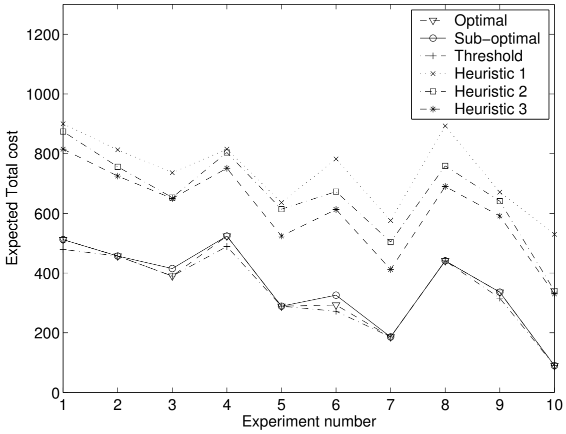


Figure 56: Total Expected Cost.

the LSPs such that there is over-dimensioning. Thus, the LSPs are re-dimensioned, when necessary, to $\Delta\%$ ($\Delta > 100$) of their capacity. These three heuristics are simple to implement and are currently in use by network operators.

First, the total expected cost is compared in Figure 56. As expected, the cost for the heuristics are much higher when compared to the proposed policies. Heuristic 1 has a lot of wasted bandwidth whereas heuristic 2 has very high signaling costs. Heuristic 3 also has a higher cost due to the combination of wasted bandwidth and frequent bandwidth adjustments. On the other hand, the three proposed policies have comparatively lower costs. Among the three policies, the sub-optimal policy has a higher cost than the optimal policy as expected. However, the threshold-based policy has an even lower cost than the optimal policy. This results from the removal of the limitation that an intermediate λSP can be used only for LSP with the same end-points. By removing this limitation, the number of λSPs to be created reduces and thus the total cost reduces. Note that this limitation can only be removed in centralized network operation because information is required about the state of the intermediate λSPs which is not available in the decentralized scenario. Thus, the cost reduction has been obtained at the expense of the added management effort.

A metric that is reflective of the operations of the policies is the number of modifications to the LSPs and λSPs. In Figure 57, the number of modifications for LSPs is shown.

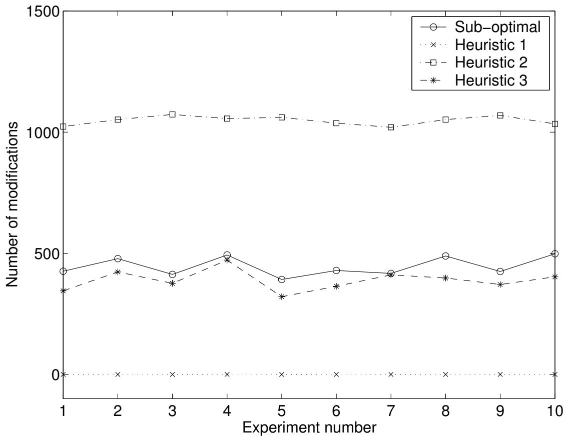
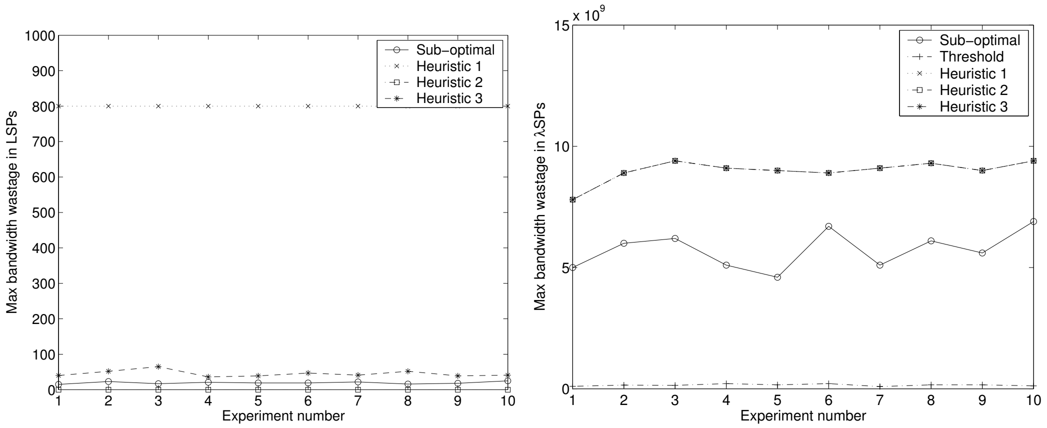


Figure 57: Number of Modifications in the LSP Topology.



(a) MPLS Network

(b) Optical Network

Figure 58: Maximum Bandwidth Wastage.

Another metric for comparison is the bandwidth wastage in the LSP and the λSP. The average and maximum wastage can be compared. The average wastage considers all the LSPs or λSPs whereas the max value gives a worst-case scenario picture of the bandwidth wastage. In Figure 58(a) and (b), the max bandwidth wastage is shown for the LSPs and λSPs, respectively. As is noticeable, the heuristics perform worse than the proposed policies. In Figure 58(a), the results for Heuristic 1 have been clipped at 800 to show the results for other policies at a reasonable scale. The actual results for Heuristic 1 are very large which show the inefficiency of the heuristic.

References

[1] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: An Overview*, IETF RFC 1633, June 1994.

[2] C. Partridge, *A Proposed Flow Specification*, IETF RFC 1363, September 1992.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource reSerVation Protocol (RSVP) Version 1 Functional Specification*, IETF RFC 2205, September 1997.

[4] S. Shenker, C. Partridge, and R. Guerin, *Specification of Guaranteed Quality of Service*, IETF RFC 2212, September 1997.

[5] J. Wroclawski, *Specification of the Controlled-Load Network Element Service*, IETF RFC 2211, September 1997.

[6] ——, *The Use of RSVP with IETF Integrated Services*, IETF RFC 2210, September 1997.

[7] P. White, "RSVP and Integrated Services in the Internet: A Tutorial," *IEEE Communications Magazine*, vol. 35, pp. 100–106, May 1997.

[8] P. White and J. Crowcroft, "The Integrated Services in the Internet: State of the Art," *Proceedings of the IEEE*, vol. 85, no. 12, pp. 1934–1946, December 1997.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services*, IETF RFC 2475, December 1998.

[10] K. Nichols, S. Blake, F. Baker, and D. Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, IETF RFC 2474, December 1998.

[11] K. Nichols, V. Jacobson, and L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, IETF RFC 2638, July 1999.

[12] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, *Assured Forwarding PHB Group*, IETF RFC 2597, June 1999.

[13] V. Jacobson, K. Nichols, and K. Poduri, *An Expedited Forwarding PHB*, IETF RFC 2598, June 1999.

[14] B. Carpenter and K. Nichols, "Differentiated services in the Internet," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1479–1494, September 2002.

[15] Y. Bernet *et al.*, *A Framework for Integrated Services Operation over Diffserv Networks*, IETF RFC 2998, November 2000.

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

[17] D. O. Awduche and B. Jabbari, "Internet traffic engineering using multi-protocol label switching (MPLS)," *Computer Networks*, vol. 40, no. 1, pp. 111–129, September 2002.

[18] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, *Overview and Principles of Internet Traffic Engineering*, IETF RFC 3272, May 2002.

[19] J. Boyle, V. Gill, A. Hannan, D. Cooper, D. Awduche, B. Christian, and W. S. Lai, *Applicability Statement for Traffic Engineering with MPLS*, IETF RFC 3346, August 2002.

[20] D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, *Requirements for Traffic Engineering over MPLS*, IETF RFC 2702, September 1999.

[21] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*. Morgan Kaufmann, 2000.

[22] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WANs," *IEEE Transactions on Communications*, vol. 40, pp. 1171–1182, July 1992.

[23] E. Mannie, P. Ashwood-Smith, D. O. Awduche, A. Banerjee, D. Basak, L. Berger, T. D. Nadeau, L. Ong, and D. P. and, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," *IETF Internet Draft, draft-ietf-ccamp-gmpls-architecture-07.txt*, May 2003, work in progress.

[24] F. L. Faucheur and W. Lai, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*, IETF RFC 3564, July 2003.

[25] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*, IETF RFC 3270, May 2002.

[26] D. Katz, K. Kompella, and D. Yeung, *Traffic Engineering (TE) Extensions to OSPF Version 2*, IETF RFC 3630, September 2003.

[27] D. O. Awduche *et al.*, *RSVP-TE: Extensions to RSVP for LSP Tunnels*, IETF RFC 3209, December 2001.

[28] P. Aukia, M. Kodialam, P. V. N. Koppol, T. V. Lakshman, H. Sarin, and B. Suter, "RATES: A Server for MPLS Traffic Engineering," *IEEE Network*, vol. 14, pp. 34–41, March/April 2000.

[29] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering," in *Proceedings of IEEE INFOCOM'01*, Anchorage, USA, April 2001, pp. 1300–1309.

[30] P. Trimintzios, L. Georgiadis, G. Pavlou, D. Griffin, C. F. Cavalcanti, P. Georgatsos, and C. Jacquenet, “Engineering the Multi-Service Internet: MPLS and IP-Based Techniques,” in *Proceedings of IEEE ICT’01*, Bucharest, Romania, June 2001.

[31] E. Mykoniati, C. Charalampous, P. Georgatsos, T. Damilatis, D. Goderis, P. Trimintzios, G. Pavlou, and D. Griffin, “Admission Control for Providing QoS in IP DiffServ Networks: the TEQUILA Approach,” *IEEE Communications Magazine*, vol. 41, no. 1, pp. 38–46, January 2003.

[32] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, “Traffic Engineering with MPLS in the Internet,” *IEEE Network Magazine*, vol. 14, no. 2, pp. 28–33, March 2000.

[33] X. Xiao, T. Telkamp, V. Fineberg, C. Chen, and L. M. Ni, “A Practical Approach for Providing QoS in the Internet Backbone,” *IEEE Communications Magazine*, vol. 40, pp. 56–62, December 2002.

[34] S. Uhlig and O. Bonaventure, “On the Cost of Using MPLS for Interdomain Traffic,” in *Proceedings of QoFIS’00*, Berlin, Germany, September 2000, pp. 141–152.

[35] D. Ooms *et al.*, *Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment*, IETF RFC 3353, August 2002.

[36] S. Chen and K. Nahrstedt, “An Overview of Quality of Service Routing for the Next Generation High Speed Networks: Problems and Solutions,” *IEEE Network*, vol. 12, no. 6, pp. 64–79, 1998.

[37] E. W. Dijkstra, “A Note on Two Problems in Connexion With Graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[38] S. Chen and K. Nahrstedt, “On Finding Multi-Constrained Paths,” in *Proceedings of ICC’98*, Atlanta, USA, June 1998, pp. 874–879.

[39] G. Feng, C. Douligeris, K. Makki, and N. Pissinou, “Performance Evaluation of Delay-Constrained Least-Cost QoS Routing Algorithms Based on Linear and Nonlinear Lagrange Relaxation,” in *Proceedings of IEEE ICC’02*, New York, USA, April 2002, pp. 2273–2278.

[40] D. S. Reeves and H. F. Salama, “A Distributed Algorithm for Delay-constrained Unicast Routing,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 239–250, April 2000.

[41] X. Yuan, “Heuristic Algorithms for Multi-Constrained Quality of Service Routing,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 244–256, April 2002.

[42] A. Jüttner, B. Szviatovszki, I. Mécs, and Z. Rajkó, “Lagrange Relaxation Based Method for the QoS Routing Problem,” in *Proceedings of IEEE INFOCOM’01*, Anchorage, USA, April 2001, pp. 859–868.

[43] T. Korkmaz and M. Krunz, “Multi-constrained Optimal Path Selection,” in *Proceedings of IEEE INFOCOM’01*, Anchorage, USA, April 2001, pp. 834–843.

[44] K. Kar, M. Kodialam, and T. V. Lakshman, “Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Application,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, December 2000.

[45] S. Suri, M. Waldvogel, and P. Warkhede, “Profile-Based Routing: A New Framework for MPLS Traffic Engineering,” in *Proceedings of 2 International Workshop on Quality of future Internet Services (QofIS’01)*, Coimbra, Portugal, September 2001, pp. 138–157.

[46] K. Lai and M. Baker, “Measuring Bandwidth,” in *Proceedings of IEEE INFOCOM’99*, New York, USA, March 1999, pp. 235–245.

[47] V. Jacobson, *Pathchar*, ftp://ftp.ee.lbl.gov/pathchar/, 1997.

[48] S. Keshav, “A Control-Theoretic Approach to Flow Control,” in *Proceedings of ACM SIGCOMM’91*, Zurich, Switzerland, September 1991, pp. 3–15.

[49] K. Lai and M. Baker, “Nettimer: A Tool for Measuring Bottleneck Link Bandwidth,” in *Proceedings of 3rd USENIX Symposium on Internet Technologies and Systems*, San Francisco, USA, March 2001, pp. 184–193.

[50] R. L. Carter and M. E. Crovella, “Measuring Bottleneck Link Speeds in Packet-Switched Networks,” *Boston University Technical Report BU-CS-96-006*, 1996.

[51] V. Paxson, “End-to-end Internet Packet Dynamics,” in *Proceedings of ACM SIGCOMM’97*, Cannes, France, September 1997, pp. 139–152.

[52] M. Jain and C. Dovrolis, “Pathload: A Measurement Tool for End-to-End Available Bandwidth,” in *Proceedings of PAM’02*, Fort Collins, USA, March 2002.

[53] S. Banerjee and A. Agarwala, “Estimating Available Capacity of a Network Connection,” in *Proceedings of IEEE ICON’00*, Singapore, Singapore, September 2000, pp. 131–138.

[54] *http://dast.nlanr.net/Projects/Iperf/*, Iperf Tool.

[55] *http://www.cisco.com/warp/public/732/Tech/netflow/*, Cisco Netflow.

[56] R. L. Carter and M. E. Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks,” *Performance Evaluation*, vol. 27,28, pp. 297–318, 1996.

[57] G. Jin, G. Yang, B. Crowley, and D. Agarwal, “Network Characterization Service (NCS),” in *Proceedings of 10th IEEE Symposium on High Performance Distributed Computing*, August 2001.

[58] C. Dovrolis, P. Ramanathan, and D. Moore, “What do Packet Dispersion Techniques Measure?” in *Proceedings of IEEE INFOCOM’01*, Anchorage, USA, April 2001, pp. 905–914.

[59] V. Robeiro *et al.*, “Multifractal Cross-Traffic Estimation,” in *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 200o.

[60] M. Jain and C. Dovrolis, “End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” in *Proceedings of ACM SIGCOMM’02*, Pittsburgh, USA, August 2002, pp. 295–308.

[61] W. C. Feng, F. Jahanian, and S. Sechrest, “An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video,” *ACM/Springer Verlag Multimedia Systems Journal*, vol. 5, no. 5, 1997.

[62] A. Adas, “Supporting Real-time VBR Video Using Dynamic Reservation Based on Linear Prediction,” in *Proceedings of IEEE INFOCOM’96*, San Francisco, USA, March 1996, pp. 1476–1483.

[63] H. Zhang and E. W. Knightly, “RED-VBR: A Renegotiation Based Approach to Support Delay Sensitive VBR Video,” *Multimedia Systems*, vol. 5, pp. 164–176, 1997.

[64] D. Reininger, G. Ramamurthy, and D. Raychaudhuri, “VBR MPEG Video Coding with Dynamic Bandwidth Renegotiation,” in *Proceedings of IEEE ICC’95*, Seattle, USA, June 1995, pp. 1773–1777.

[65] A. Terzis, L. Wang, J. Ogawa, and L. Zhang, “A Two-Tier Resource Management Model for the Internet,” in *Proceedings of IEEE GLOBECOM’99*, Rio de Janeiro, Brazil, December 1999, pp. 1779–1791.

[66] P. S. Maybeck, *Stochastic Models, Estimation, and Control.* Academic Press, Inc, 1979.

[67] A. Terzis, “A two-tier resource allocation framework for the internet,” Ph.D. dissertation, University of California, Los Angeles, 2000.

[68] C. N. Chuah, L. Subramanian, R. H. Katz, and A. D. Joseph, “QoS Provisioning Using a Clearing House Architecture,” in *Proceedings of 8th International Workshop on Quality of Service*, Pittsburgh, USA, June 2000, pp. 115–124.

[69] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, “A Flexible Model for Resource Management in Virtual Private Networks,” in *Proceedings of ACM SIGCOMM’99*, Cambridge, USA, September 1999, pp. 95–108.

[70] *White paper: Cisco MPLS AutoBandwidth Allocator for MPLS Traffic Engineering*, http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/mpatb_wp.pdf, June 2001.

[71] D. Banerjee and B. Mukherjee, “Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs, and a Reconfiguration Study,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 598–607, 2000.

[72] R. M. Krishnaswamy and K. N. Sivarajan, “Design of Logical Topologies: a Linear Formulation for Wavelength-routed Optical Networks with no Wavelength Changers,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, 2001.

[73] M. A. Marsan, A. Bianco, E. Leonardi, and F. Neri, “Topologies for Wavelength-routing All-optical Networks,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 5, 1993.

[74] R. Dutta and G. N. Rouskas, “A Survey of Virtual Topology Design Algorithms for Wavelength Routed Optical Networks,” North Carolina State University, Tech. Rep. TR-99-06, 12 1999.

[75] A. Gençata and B. Mukherjee, “Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 236–247, April 2003.

[76] J. C. de Oliveira, C. Scoglio, T. Anjali, L. C. Chen, I. F. Akyildiz, J. A. Smith, G. Uhl, and A. Sciuto, “Design and Management Tools for an MPLS Domain QoS Manager,” in *Proceedings of SPIE ITCOM’02*, Boston, USA, July 2002, pp. 43–54.

[77] I. F. Akyildiz, T. Anjali, L. C. Chen, J. C. de Oliveira, C. Scoglio, A. Sciuto, J. A. Smith, and G. Uhl, “A New Traffic Engineering Manager for DiffServ/MPLS Networks: Design and Implementation on an IP QoS Testbed,” *Computer Communications*, vol. 26, no. 4, pp. 388–403, March 2003.

[78] J. Case, M. Fedor, M. Schoffstall, and J. Davin, *A Simple Network Management Protocol (SNMP)*, IETF RFC 1157, May 1990.

[79] W. Stallings, *SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards*. Addison-Wesley, 1993.

[80] K. McCloghrie and M. T. Rose, *Management Information Base for network management of TCP/IP-based internets*, IETF RFC 1156, May 1990.

[81] R. Callon, “Predictions for the Core of the Network,” *IEEE Internet Computing*, vol. 4, no. 1, pp. 60–61, January/February 2000.

[82] J. C. de Oliveira, T. Anjali, B. King, and C. Scoglio, “Building an IP Differentiated Services Testbed,” in *Proceedings of IEEE ICT’01*, Bucharest, Romania, June 2001.

[83] T. Ye, D. Harrison, B. Mo, B. Sikdar, H. T. Kaur, S. Kalyanaraman, B. Szymanski, and K. Vastola, “Traffic Management and Network Control Using Collaborative On-line Simulation,” in *Proceedings of IEEE ICC’01*, Helsinki, Finland, June 2001, pp. 204–209.

[84] Y. Yemini, A. V. Konstantinou, and D. Florissi, “NESTOR: An Architecture for Network Self-Management and Organization,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 5, pp. 758–766, May 2000.

[85] C. Tsarouchis *et al.*, “A Policy-Based Management Architecture for Active and Programmable Networks,” *IEEE Network*, vol. 17, pp. 22–28, May/June 2003.

[86] J. C. de Oliveira, “New Techniques for End-to-end Quality of Service Provisioning in DiffServ/MPLS Networks,” Ph.D. dissertation, Georgia Institute of Technology, 2003.

[87] C. Scoglio, T. Anjali, J. C. de Oliveira, I. F. Akyildiz, and G. Uhl, “A New Threshold-Based Policy for Label Switched Path Setup in MPLS Networks,” in *Proceedings of 17th ITC’01*, Salvador, Brazil, September 2001, pp. 1–12.

[88] T. Anjali, C. Scoglio, J. C. de Oliveira, I. F. Akyildiz, and G. Uhl, “Optimal Policy for LSP Setup in MPLS Networks,” *Computer Networks*, vol. 39, no. 2, pp. 165–183, June 2002.

[89] H. Saito, Y. Miyao, and M. Yoshida, “Traffic Engineering Using Multiple Multipoint-to-Point LSPs,” in *Proceedings of IEEE INFOCOM’00*, Tel Aviv, Israel, March 2000, pp. 894–901.

[90] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis, *Constraint-Based LSP Setup using LDP*, IETF RFC 3212, January 2002.

[91] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

[92] T. Anjali and C. Scoglio, “Traffic Routing in MPLS Networks based on QoS Estimation and Forecast,” *Submitted for publication*, 2004.

[93] Z. Wang and J. Crowcroft, “Quality-of-Service Routing for Supporting Multimedia Applications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1288–1234, September 1996.

[94] B. Zhang, M. Krunz, H. T. Mouftah, and C. Chen, “Stateless QoS Routing in IP Networks,” in *Proceedings of IEEE GLOBECOM’01*, San Antonio, USA, November 2001, pp. 1600–1604.

[95] D. H. Lorenz and A. Orda, “QoS Routing in Networks with Uncertain Parameters,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 768–778, December 1998.

[96] R. Guerin, “QoS Routing in Networks with Inaccurate Information: Theory and Algorithms,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, June 1999.

[97] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, “Improving QoS Routing Perofrmance Under Inaccurate Link State Information,” in *Proceedings of 16th ITC’99*, Edinburgh, UK, June 1999.

[98] T. Korkmaz and M. Krunz, “Bandwidth-Delay Constrained Path Selection Under Inaccurate State Information,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 384–398, June 2003.

[99] T. Anjali, C. Scoglio, L. Chen, I. F. Akyildiz, and G. Uhl, “ABEst: An Available Bandwidth Estimator within an Autonomous System,” in *Proceedings of IEEE GLOBECOM’02*, Taipei, Taiwan, November 2002, pp. 2360–2364.

[100] T. Anjali, C. Scoglio, I. F. Akyildiz, J. A. Smith, and A. Sciuto, “MABE: A New Method for Available Bandwidth Estimation in an MPLS Network,” in *Proceedings of IEEE Networks’02*, Atlanta, USA, August 2002, pp. 295–306.

[101] *http://www.ietf.org/html.charters/ippm-charter.html*, IP Performance Metrics Working Group.

[102] R. Guerin, A. Orda, and D. Williams, “QoS Routing Mechanisms and OSPF Extensions,” in *Proceedings of IEEE GLOBECOM’97*, Phoenix, USA, November 1997, pp. 1903–1908.

[103] J. C. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl, “A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting,” in *Proceedings of IEEE INFOCOM’02*, New York, USA, June 2002, pp. 695–704.

[104] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, *Introduction to version 2 of the Internet-standard Network Management Framework*, IETF RFC 1441, April 1993.

[105] K. McCloghrie and M. Rose, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, IETF RFC 1213, March 1991.

[106] *http://people.ee.ethz.ch/ oetiker/webtools/mrtg/*, MRTG Website.

[107] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, 1996.

[108] F. Kelly, *Stochastic Networks: Theory and Applications*, ser. Royal Statistical Society Lecture Notes Series, F. P. Kelly, S. Zachary, and I. B. Ziedins, Eds. 4. Oxford University Press, 1996.

[109] J. Yang and M. Devetsikiotis, “On-line Estimation, Network Design and Performance Analysis with Effective Bandwidths,” in *Proceedings of 17th ITC’01*, Salvador, Brazil, September 2001, pp. 347–358.

[110] *http://abilene.internet2.edu/*, Abilene Website.

[111] T. Anjali and C. Scoglio, “TEMB: Tool for End-to-End Measurement of Available Bandwidth,” in *Proceedings of IEEE ELMAR’03*, Zadar, Croatia, June 2003.

[112] T. Anjali, C. Scoglio, and G. Uhl, “A New Scheme for Traffic Estimation and Resource Allocation for Bandwidth Brokers,” *Computer Networks*, vol. 41, no. 6, pp. 761–777, April 2003.

[113] T. Anjali, C. Bruni, D. Iacoviello, G. Koch, C. Scoglio, and S. Vergari, “Optimal Filtering in Traffic Estimation for Bandwidth Brokers,” in *Proceedings of IEEE GLOBECOM’03*, San Franciso, USA, December 2003, pp. 3636–3640.

[114] B. Teitelbaum and P. F. Chimento, *QBone BB Architecture*, Internet2 Document, http://qbone.internet2.edu/bb/bboutline2.html, June 2000.

[115] J. W. Roberts, “Traffic Theory and the Internet,” *IEEE Communications Magazine*, vol. 39, no. 1, pp. 94–99, January 2001.

[116] A. Kolarov, A. Atai, and J. Hui, “Application of Kalman Filter in High-Speed Networks,” in *Proceedings of IEEE GLOBECOM’94*, San Francisco, USA, November 1994, pp. 624–628.

[117] T. Bonald, S. Oueslati-Boulahia, and J. Roberts, “IP-traffic and QoS Control: Towards a Flow-aware Architecture,” in *Proceedings of 18th World Telecommunication Congress*, Paris, France, 2002.

[118] B. Cipra, “Engineers Look to Kalman Filtering for Guidance,” *SIAM News*, vol. 26, no. 5, 1993.

[119] L. Kleinrock, *Queueing Systems*. John Wiley, 1975.

[120] C. Scoglio, T. Anjali, J. C. de Oliveira, L. Chen, I. F. Akyildiz, G. Uhl, and J. A. Smith, “TEAM: A Traffic Engineering Automated Manager for DiffServ-based MPLS Networks,” *submitted for publication*, 2004.

[121] V. Jimenez and A. Marzal, “Computing the K Shortest Paths: A New Algorithm and an Experimental Comparison,” *Lecture Notes in Computer Science, Springer Verlag*, vol. 1668, pp. 15–29, 1999.

[122] T. Anjali, C. Scoglio, and I. F. Akyildiz, “LSP and λSP Setup in GMPLS Networks,” in *Proceedings of IEEE INFOCOM’04*, Hong Kong, China, March 2004.

[123] A. Banerjee, J. Drake, J. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, “Generalizd Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques,” *IEEE Communications Magazine*, vol. 39, no. 7, pp. 144–151, July 2001.

[124] R. D. Davis, K. Kumaran, G. Liu, and I. Sainee, “SPIDER: A Simple and Flexible Tool for Design and Provisioning of Protected Lightpaths in Optical Networks,” *Bell Labs Technical Journal*, pp. 82–97, January-June 2001.

[125] C. Chen and S. Banerjee, “Optical Switch Configuration and Lightpath Assignment in Wavelength Routing Multihop Lightwave Networks,” in *Proceedings of IEEE INFOCOM’95*, Boston, USA, April 1995, pp. 1300–1307.

Vita

Tricha Anjali was born in Lucknow, India on July 22, 1976. She completed her schooling at Hardwar, India after which she joined the Indian Institute of Technology, Bombay (Mumbai). She received her Integrated Master of Technology (M. Tech.) in Electrical Engineering from the IIT in May 1998. During this period, she was funded by the National Talent Search scholarship from the Government of India. In the Spring of 2000, she joined the doctoral program in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta. She is a member of IEEE communications society and the ACM. She received the Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in April 2004. Her research interests lie in computer and broadband networks, network management and network measurement.