Preemption and re-routing control strategy for resource provisioning and path selection in connection oriented networks

Lau, Chun Hau

2008

Lau, C. H. (2008). Preemption and re-routing control strategy for resource provisioning and path selection in connection oriented networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4120>

<https://doi.org/10.32657/10356/4120>

Nanyang Technological University

Downloaded on 26 Apr 2023 16:46:42 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**PREEMPTION AND RE-ROUTING CONTROL
STRATEGY FOR RESOURCE PROVISIONING
AND PATH SELECTION IN CONNECTION
ORIENTED NETWORKS**

**LAU CHUN HAU
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING
2008**

Preemption and Re-routing Control Strategy for Resource Provisioning and Path Selection in Connection Oriented Networks

Lau Chun Hau

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

Acknowledgements

I would like to express my gratitude to my supervisor and mentor, Associate Professor Soong Boon Hee for his guidance. He has great vision and helps me to shape the direction of this research work. I would also like to express my gratitude to Associate Professor Sanjay Kumar Bose for his invaluable advice. He has given me great support especially when the research seems to encounter stalemate at times. For part of the thesis, the author used an excellent software package, SPNP (Stochastic Petri Net Package) from Duke University, for analyzing Stochastic Petri Nets. It is an indispensable tool and the author thanks Prof. Kishor. S. Trivedi for the use of the software.

I am also grateful to Nanyang Technological University for the financial support, which has enabled me to pursue this research. Finally, I would also like to express my thanks to my family and my friends.

Table of Contents

Acknowledgements.....	i
Table of Contents	ii
Summary	v
List of Figures.....	vii
List of Tables	xi
List of Abbreviations	xii
1. Introduction	1
1.1 MOTIVATION	1
1.2 OBJECTIVES.....	2
1.3 MAJOR CONTRIBUTIONS OF THE THESIS.....	3
1.4 ORGANIZATION OF THE THESIS	4
2. Literature Overview	6
2.1 INTEGRATED SERVICES	7
2.2 DIFFERENTIATED SERVICES.....	9
2.3 MPLS NETWORKS	13
2.4 MPLS TRAFFIC ENGINEERING	15
2.5 RESOURCE ALLOCATION AND PREEMPTION	19
2.5.1 Admission Control.....	21
2.5.2 Resource Reservation.....	22
2.5.3 Routing.....	24
2.5.4 Path Protection	27

2.5.5 Preemption	28
2.5.5.1 Preemption in ATM	29
2.5.5.2 Preemption in Wireless Networks.....	32
2.5.5.3 Preemption in MPLS.....	33
2.6 SUMMARY	35
3. Preemption with Re-routing to Minimize Service Disruption	38
3.1 PROBLEM FORMULATION	40
3.2 PREEMPTION FRAMEWORK.....	43
3.3 HEURISTICS PREEMPTION ALGORITHM	48
3.4 CENTRALIZED PREEMPTION STRATEGY.....	56
3.4.1 Computational Complexity	62
3.5 DECENTRALIZED PREEMPTION STRATEGY	62
3.6 SIMULATION RESULTS AND DISCUSSIONS	66
3.7 SIMULATIONS ON NON-UNIFORM LINKS	73
3.8 SUMMARY	75
4. Path Selection with Preemption and Re-routing Control for MPLS	77
4.1 NETWORK MODEL.....	78
4.2 PREEMPTION AND RE-ROUTING STRATEGY	81
4.2.1 Search All and Preempt (SEP).....	82
4.2.2 Limit Hop Count and Preempt (LIP)	83
4.2.3 Re-routing Control.....	84
4.2.4 Congestion Based Preemption Algorithm.....	86
4.3 PERFORMANCE EVALUATION	88
4.4 DECENTRALIZED PREEMPTION STRATEGY	100
4.4.1 Problem 4.1: Maximum Flow for Local Re-route.....	103
4.4.2 Problem 4.2: Routing to Minimize the Network Load	105
4.4.3 Problem 4.3: Multiple Re-routes.....	107
4.5 PERFORMANCE COMPARISONS	108
4.6 SUMMARY	113

5. Stochastic Petri Net Model for Preemption	115
5.1 MOTIVATION	115
5.2 BRIEF DESCRIPTION ON STOCHASTIC PETRI NET	116
5.2.1 <i>Stochastic Petri Net Package (SPNP) Modeling Tool</i>	117
5.3 SERA PREEMPTION ALGORITHM	118
5.3.1 <i>Formulation of SERA</i>	118
5.3.2 <i>Properties of SERA</i>	122
5.3.3 <i>Non-Preemption SPN Model</i>	126
5.3.4 <i>Preemption SPN Model</i>	128
5.3.5 <i>Numerical Results</i>	129
5.4 NETWORK PREEMPTION SPN MODEL	133
5.4.1 <i>Preemption Strategy</i>	133
5.4.2 <i>Network Preemption Model</i>	135
5.4.2.1 <i>New Arrival Subnet</i>	136
5.4.2.2 <i>Link Subnet</i>	140
5.4.2.3 <i>Connection Depository Subnet</i>	141
5.4.3 <i>Numerical Results</i>	144
5.5 SUMMARY	153
6. Conclusion and Recommendations.....	154
6.1 CONCLUSION	154
6.2 RECOMMENDATIONS FOR FUTURE RESEARCH.....	156
Author's Publications	157
Bibliography	158
Appendix I	168

Summary

This thesis focuses on the preemption issues in connection-oriented networks, e.g. MPLS and ATM. Preemption mechanism primarily resides in the network management realm, where it functions by tearing down existing lower priority connections in order to provide resources for the higher priority connections. Generally, it is activated when the network experiences high congestion due to competitions between different connections and network resources must be provided to satisfy the new mission critical connections. However, by giving these limited resources to the critical connections through preemption, the duration spent and work done for the preempted connections are wasted. This could lead to lower network throughput as well as higher service interruption. Our first objective is to minimize this service disruption and loss of throughput. The general strategy is to provide re-routing capability to the existing connections, such that, these connections are re-routed before being torn down. By re-routing the connections as far as possible, the author shows that the network throughput loss is minimized while the service disruption is reduced significantly. Subsequently, this strategy is extended by using the preemption mechanism as an active entity for network path selection. Instead of using a longer alternative path (e.g. higher hop counts) when the network is congested, higher priority connections are allowed to route on the shortest possible path even though preemption may be triggered. It is shown that this strategy greatly improves network throughput as compared to existing preemption schemes since the

total network resource consumption is minimized. Finally, given the myriads of existing preemption schemes, a common platform based on Stochastic Petri Nets is provided for preemption study purposes. Two models are proposed in this framework, i.e. single link model and network model. Analytic-numeric results indicate that the models are able to give important metrics of the preemption performances. Future study will focus on the stability issues of preemption.

List of Figures

Fig. 2.1: Per-flow Reservation between a Sender and a Receiver.	8
Fig. 2.2: DSCP Defined in the TOS Field of IPv4 Header.	10
Fig. 2.3: DSCP Defined in the Traffic Class Field of IPv6 Header.	10
Fig. 2.4: A Differentiated Services Model.	12
Fig. 2.5: Label Switching in MPLS.	14
Fig. 2.6: Traffic Engineering Processes.	16
Fig. 2.7: LSP Routed on Not Over-utilized Path to Avoid Congestion.	17
Fig. 2.8: Resource Reservation in Multiple Domains Using Bandwidth Broker.	23
Fig. 2.9: Global Re-route.	27
Fig. 2.10: Local Re-route.	27
Fig. 2.11: Preemption Processes.	29
Fig. 3.1: The Routine to Obtain the Set of Preemptable Connections.	44
Fig. 3.2: OP-Prio Preemption Routine.	47
Fig. 3.3: Performance Comparisons of Different Preemption Strategies.	48
Fig. 3.4: Modified Heuristics Algorithm (MHA).	49
Fig. 3.5: Performances of OP-Prio, Heuristics and MHA.	51
Fig. 3.6: Normalized Throughput for Single Link Scenario.	52
Fig. 3.7: Probability of Success for Single Link Scenario.	53
Fig. 3.8: Probability of Success for Priority 0 Connections.	55
Fig. 3.9: Probability of Success for Priority 7 Connections.	55

Fig. 3.10: Construction of Subgraph.	58
Fig. 3.11: Local Re-routing by the Congested Link.	64
Fig. 3.12: Network Topologies.	66
Fig. 3.13: Network Normalized Throughput.	68
Fig. 3.14: Probability of Success.	69
Fig. 3.15: Service Disruption Probability.	70
Fig. 3.16: Probability of Re-routing.	71
Fig. 3.17: Percentage of Call Completion for Decentralized Strategy.	71
Fig. 3.18: Percentage of Call Completion for Decen. Strategy with No Re-routing.	73
Fig. 3.19 : Network Topology with Non-uniform Links.	74
Fig. 3.20 : Normalized Throughput for Non-Uniform Links.	74
Fig. 3.21 : Service Disruption Probability for Non-Uniform Links.	75
Fig. 4.1: Source-destination Node Pair and Possible Paths.	79
Fig. 4.2: An Illustration of SEP Algorithm.	83
Fig. 4.3: An Illustration of LIP Algorithm.	84
Fig. 4.4: An Illustration of Re-routing Control.	86
Fig. 4.5: NFS-NET Network Topology.	88
Fig. 4.6: Throughput for No Preemption, PREM and SEP without Re-routing.	89
Fig. 4.7: Probability of Success for Prio 0 LSPs without Re-routing.	91
Fig. 4.8: The Effects of σ on Network Throughput and Re-routing Probability.	91
Fig. 4.9: Network Throughput of LIP, SEP and No Preemption.	93
Fig. 4.10: Re-routing Probability of LIP (Case A, Case B, Case C) and SEP.	94
Fig. 4.11: Probability of Success for LIP (Case A, Case B, Case C) and SEP.	95
Fig. 4.12: Average Path Length of LIP, SEP and No Preemption.	96
Fig. 4.13: Distribution of LSPs Completed Successfully for No Preemption.	97

Fig. 4.14: Distribution of LSPs Completed Successfully for LIP Case B.	98
Fig. 4.15: Reduction of Overall Traffic Load by Local-Rerouting.	101
Fig. 4.16: The Flow Diagram of the Decentralized Strategy.	108
Fig. 4.17: Throughput Comparison of Various Preemption Strategies.	109
Fig. 4.18: Re-routing Probability of SEP, LIP and Decentralized Strategies.	110
Fig. 4.19: Preemption Probability for SEP, LIP and Decentralized Strategies.	111
Fig. 4.20: Average Path Length for No Preemption, SEP, LIP and Decen. Stra.	111
Fig. 5.1. Basic Elements of SPN.	116
Fig. 5.2: SPNP Modeling Tool.	117
Fig. 5.3: Bandwidth and Holding Time of Connections.	119
Fig. 5.4: Subsequence of Arriving Connections.	123
Fig. 5.5: Non-Preemption SPN Model.	126
Fig. 5.6 Preemption SPN Model.	128
Fig. 5.7: Link Useful Utilization.	130
Fig. 5.8: Blocking Probability.	132
Fig. 5.9: Preemption Probability	132
Fig. 5.10: New Arrival Subnet.	136
Fig. 5.11: <i>ADMIT</i> Guard Function.	139
Fig. 5.12: <i>DENY</i> Guard Function.	139
Fig. 5.13: <i>PREEMPT_ALERT</i> Guard Function.	139
Fig. 5.14: Link Subnet	140
Fig. 5.15: <i>DRAW_BW</i> Guard Function.	141
Fig. 5.16: <i>RETURN_BW</i> Guard Function	141
Fig. 5.17: Connection Depository Subnet.	141
Fig. 5.18: <i>REAL_PREEMPT</i> Guard Function.	144

List of Figures	x
<hr/>	
Fig. 5.19: Network Topologies for SPN Network Model.	145
Fig. 5.20: Numerical and Simulation Results on Network Throughput.	148
Fig. 5.21: Network Throughput.	149
Fig. 5.22: Preemption Probability.	150
Fig. 5.23: Network Throughput for ‘Priority’ Case in the 5-Node Topology.	152
Fig. 5.24: Preemption Probability for ‘Priority’ Case in the 5-Node Topology.	152

List of Tables

Table 2.1: A List of Existing Preemption Schemes.	35
Table 3.1: Bandwidth and Priority Information of 16 Connections on Single Link.	47
Table 5.1: Immediate Transitions Firing Priority for Single Link Model.	129
Table 5.1: Sizes of the Markov Chains for the Single Link Model.	130
Table 5.3: Immediate Transitions Firing Priority for Network Model.	144
Table 5.4: Node Pairs and Links Used for 4-Node Topology.	147
Table 5.5: Node Pairs and Links Used for 5-Node Topology.	147
Table 5.6: Sizes of the Markov Chains for the Network Model.	147

List of Abbreviations

AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BGP	Border Gateway Protocol
CLS	Controlled-Load Service
DAMA	Demand Assigned Multiple Access
DSCP	Differentiated Services Code Points
EF	Expedited Forwarding
GS	Guaranteed Service
LER	Label Edge Router
LG	Loss Bound Guaranteed
LSP	Label Switch Path
LSR	Label Switch Router
MBAC	Measurement-based Admission Control
MPLS	Multi-protocol Label Switching
OSPF	Open Shortest Path First
PBAC	Parameter-based Admission Control
PHB	Per-Hop Behavior
QoS	Quality of Service
RSVP	Reservation Protocol
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SONET	Synchronous Optical Net
SPN	Stochastic Petri Nets
TE	Traffic Engineering
VoIP	Voice over IP
WATM	Wireless ATM

1. Introduction

1.1 Motivation

Given a fixed network with links that have limited resources and subject to increasing network load, there exists a point in time when the network could no longer admit new connections. Under such a network congestion scenario, new connections will be rejected regardless of its priority levels. In order to provide resources to the high priority connections, preemption mechanism is used to acquire network resources from the existing lower priority connections. Connection preemption is defined as the process of tearing down one or more existing lower priority connections in order to provide sufficient and reliable services to a higher priority new request. Different objectives can be used to select the combination of connections for preemption, e.g. bandwidth minimization. However, by tearing down existing connections, the ongoing services will be disrupted. This will affect network throughput and end users' perception on the network services provided.

The main motivation to this thesis is to design a preemption strategy that will minimize service disruption to end users. Most of the current strategies focus on different preemption criteria without taking into consideration the re-routing capability. For example, a scheme may be designed to minimize the number of connections preempted. However, re-routing capability can fundamentally influence the ways connections are selected. Instead of minimizing the number of connections

preempted, the network may preempt connections that can be re-routed so that the ongoing services are not disrupted.

Furthermore, the author is interested in using the preemption mechanism as an integrated element in connection routing process. By implementing preemption, a network is able to route a higher priority connection on a more favorable path. These strategies effectively move the preemption mechanism into the active network management domain, in which its main role is to reorganize the network connection such that higher priority connections can be serviced without detrimentally disrupting lower priority connections.

1.2 Objectives

The first objective of this work is to minimize service disruption caused by the preemption of ongoing connections. The author implements re-routing in such ways that the connections that are to be preempted are re-routed as far as possible. This ensures that end users of the re-routed connections will not experience service disruption due to the underlying preemption process. Equipped with this functionality, the author seeks to minimize service disruption by selecting the re-routable connections for preemption.

The second objective is to investigate the effects of preemption mechanism in the active network management domain. The author implements preemption side-by-side with routing algorithm in order to search for the feasible path. The routing algorithm has the option to trigger preemption to route a connection on a more favorable path, even though other feasible paths may exist. This fundamentally

differs from the traditional approach where all the feasible paths are exhausted before triggering preemption.

The third objective is to formulate an analytic-numeric model to evaluate the network performance of preemption. The model is developed based on the Stochastic Petri Nets (SPN). It can be used to validate the preemption strategy designed as well as for performance comparisons between different strategies.

1.3 Major Contributions of the Thesis

The major contributions of this thesis are as follows:

1. The development of preemption strategy with re-routing to minimize service disruption. Instead of using criteria such as priority and bandwidth, the algorithm is designed to select the re-routable connections. Service disruption is significantly reduced as compared to existing strategies.
2. We formulate a preemption strategy that is based on connection service time. The objective is to minimize the loss of network throughput, which is given by the product of bandwidth and service time. As resources committed to the newly started connections are minimal, preempting them will not deeply affect the network throughput. An approximation method is developed to statistically estimate the connection service time if the exact value is not given in advance.
3. The advantages of using preemption in conjunction with routing are investigated. Whenever a network is congested, the routing algorithm may have to exhaust all the feasible paths before triggering preemption to acquire the necessary resources. If the path chosen is of greater length than the

shortest possible path, more resources are utilized. This will adversely affect future connections especially if the connections routed are of higher priority levels. By consistently routing higher priority connections on the shortest path and using preemption if necessary, the network is able to achieve higher throughput.

4. For the purpose of easier integration and implementation in the current Internet technologies, decentralized preemption strategies are developed. Instead of using a central controller, the preemption decisions are made on the individual network links based solely on the local information. These decentralized strategies expedite the decision making as well as improve scalability.
5. We develop analytic-numeric models for network preemption based on Stochastic Petri Nets (SPN). The first model is used to analyze preemption on a single link. This simple version is subsequently expanded to model network preemption. The main purpose of this work is to validate preemption strategies and compare network performance.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 presents the current state-of-the-art in the literature, reviews some of the techniques that can be used to minimize preemption events and describes the reasons why preemption is required. The chapter also examines in detail the existing preemption strategies. Chapter 3 presents the problem formulation and preemption strategy to minimize service disruption. The decentralized strategy for service disruption minimization is formulated in this

chapter. Chapter 4 details the path searching strategies based on routing and preemption. Two strategies are developed – one that triggers preemption actively and the other that exhausts all feasible paths before triggering preemption. This chapter also describes the corresponding decentralized approach.

Chapter 5 presents the SPN models for network preemption. This chapter includes a brief description on the basics of SPN. The performance of preemption on a single link as well as on different network topologies is illustrated. Finally, Chapter 6 concludes the findings in this thesis and proposes the future directions.

2. Literature Overview

This chapter presents the literature surveys of the current research work on topics related to this thesis. The surveys cover the topics of Quality of Service (QoS), Multi-protocol Label Switching (MPLS) networks, admission control, resource reservation, routing, path protection, and preemption. The research gaps that motivate this thesis will be discussed in detail later in this chapter.

The most common and simplest form of service that current networks provide is *Best Effort* service. It does not provide resource assurance to traffic flows. When a link is congested, packets are dropped indiscriminately from the queues. However, since networks treat all the Best Effort packets equally, any flow could be affected by network congestion. Although Best Effort service is adequate for some non-real time applications such as file transfer and email, it could pose problems to mission critical or high priority traffic such as Voice over IP (VoIP) and video conference, where most of the information is delay and loss-sensitive. New architectures for resource allocation that support resource assurance and different levels of services are essential for the networks to evolve into multi-service networks.

Two of the most important architectures for resource allocation are – *Integrated Services* and *Differentiated Services* [1]. Integrated Services [2] provide resource assurance through resource reservation for individual application flows, whereas Differentiated Services [3] use a combination of edge policing, marking, provisioning, and traffic prioritization. The main objective of implementing

Integrated Services and Differentiated Services is to provide QoS to the data traffic being carried by the networks. QoS can be generally referred to as the capability to provide resource assurance and service differentiation in a network [4]. It can also be described as a set of measurable parameters, such as delay, throughput, and loss rate [19]. Resource assurance defines the way the resource of a network should be allocated to packets transiting in a network. Service differentiation encompasses the ability of a network to classify the application a packet belongs to and provides corresponding supports to the packet. A network that supports QoS needs to take an active role in the resource allocation process and decides how much and who should get the resources. For instance, packets of VoIP applications should be allocated to the shortest possible paths so that the delay bound is not violated. At a queue that overflows, the link should drop loss-insensitive data such as web browsing instead of more mission critical business transaction information. QoS provided is not limited to wired networks but can be extended to wireless networks as well [20-22].

2.1 Integrated Services

The Integrated Services [2] architecture is based on per-flow resource reservation. Network flow is defined as a simplex and distinguishable stream of data that originated from a single user activity and require the same QoS [5]. To receive resource assurance, an application or network flow must make a reservation before it can transmit traffic in the network. Resource reservation involves several steps. The application must first characterize its traffic source and the resource requirements. The network then uses a routing protocol to find a path that satisfies the requested resources. Resource reservation protocol such as (RSVP) [2] is used to acquire the

requested resources along that path. At each hop admission control checks whether sufficient resources are available to accept the new reservation. Once the reservation is established, the application can start to send traffic over the path reserved. Generally, Integrated Services include the functions of resource-based admission control, packet scheduling and buffer management [6].

Two service classes – Guaranteed Service (GS) [7] and Controlled-load service (CLS) [8] are supported by Integrated Services. GS provides explicitly guaranteed network services such as upper bound end-to-end delay and loss rate. The traffic carried by GS must strictly follow the parameters specified such as peak rate, packet size, and token bucket parameters. On the other hand, CLS provides services to applications that can tolerate packet delay jitters and require no upper bound. The performance resembles closely a network that is not heavily loaded and it is more appropriate for adaptive real-time communications. Fig. 2.1 illustrates the setting up of per-flow reservation in Integrated Services. The sender transmits a PATH message to probe the network and the receiver confirms the reservation by replying a RESV message.

Fig. 2.1: Per-flow Reservation between a Sender and a Receiver.

There are several factors that hamper the deployment of Integrated Services. Although per-flow reservation can guarantee resources for long sessions such as

video conferencing, it is not appropriate for Web traffic which is characterized by short-lived transactions. The overheads for setting up a reservation for each session are very high. There are genuine concerns about the scalability of the mechanisms for supporting Integrated Services. In order to support per-flow reservation, each node in a network has to implement per-flow classification and scheduling. These mechanisms may not be able to handle a very large number of flows at high speeds. Furthermore, per-flow reservation across several network domains will remain a challenge as the service providers have to agree upon a common reservation protocol and resolve the different pricing rates involved.

However, Integrated Services architecture may become a viable framework for resource allocation in corporate networks. Corporate networks are typically limited in size and operated by a single administrative domain which will not pose any scalability issue as discussed. It can be used to support guaranteed bandwidth for IP telephony and video conferencing over corporate intranets.

2.2 Differentiated Services

The Differentiated Services [3], [9] architecture was developed as an alternative resource allocation scheme for Integrated Services. In the Differentiated Services architecture, users' traffic at the entry point to a network is conditioned and divided into a small number of forwarding classes by the edge routers. The edge routers usually perform traffic policing to protect the network from misbehaving traffic sources. Nonconforming traffic may be dropped, delayed, or marked with a different forwarding class. Therefore for each forwarding class, the amount of traffic that users can inject into the network is limited at the edge of the network. By changing the total

amount of traffic allowed in the network, service providers can adjust the level of resource provisioning and hence control the degree of resource assurance to the users. The mapping of packets to their appropriate forwarding classes is typically done based on the service level agreement (SLA) between the user and its service provider. SLA is a contract that specifies the forwarding service a customer should receive [5].

Fig. 2.2: DSCP Defined in the TOS Field of IPv4 Header.

Fig. 2.3: DSCP Defined in the Traffic Class Field of IPv6 Header.

Packets of different forwarding classes will be marked by the edge routers with different Differentiated Services Code Points (DSCP) in the IP headers [10]. Fig. 2.2 and Fig. 2.3 show the location of DSCP in the IPv4 and IPv6 headers, respectively. In IPv4 header, DSCP is defined in the Type of Service (TOS) field. In IPv6 header, DSCP is defined in the Traffic Class field. The code points will be used by the intermediate routers to provide the same Per-Hop Behavior (PHB), which is the

externally observable forwarding behavior for the aggregate packets of the same class. For example, at the network queues, packets that belong to different PHBs may be scheduled differently. Thus an external observer will notice the differences in queuing time for these aggregate packets. The two common PHBs are given by Assured Forwarding (AF) PHB [11] and Expedited Forwarding (EF) PHB [12]. AF PHB is used to provide reliable services to customers. Packets are forwarded with high probability even in the face of network congestion and this service guarantees the peak rate of users' data flows. EF PHB ensures that the aggregate packets departure rate is strictly bounded by a rate specified for it. Therefore, EF PHB can be seen as a loosely-guaranteed service for delay and jitter. It gains a relatively higher throughput than best effort traffic during periods of network congestion. In [23], Loss-bound Guaranteed (LG) service is proposed to complement AF and EF PHB. As LG PHB is designed to guarantee the loss rate of traffic flows, it is used for applications that are very sensitive of packets losses such video conferencing. A performance comparison between streaming traffic and elastic traffic over Differentiated Services architecture is given by [24].

Differentiated Services do not require resource reservation setup and the forwarding classes apply to traffic aggregates rather than to individual flows. This will essentially eliminate many of the scalability concerns with Integrated Services. The functions that network interior nodes have to perform to support Differentiated Services are relatively simple. The complex process of classification is needed only at the edge of the network, where traffic rates are typically much lower.

In [25], a framework for implementing Differentiated Services QoS guarantees in IP-based networks is presented. The framework takes into consideration of traffic

handling mechanisms such as traffic control (scheduling, policing), admission control, and resource allocation for short-term and long-term traffic. The implementation of Differentiated Services in wireless networks for supporting QoS is proposed in [26]. The issues considered include signaling requirements, mobility, losses, lower wireless bandwidth and battery power constraints. Although Differentiated Services gain higher popularity than Integrated Services due to its scalability, there are efforts to combine the advantages of both architectures in order to better satisfy QoS requested. For example, RSVP is used in Differentiated Services domain to perform the task of admission control such that the network can admit new requests without upsetting existing traffic [27].

Fig. 2.4: A Differentiated Services Model.

An example of the Differentiated Services model is given in Fig. 2.4. Admission control is used to ensure that the network is not overloaded and the arriving traffic conforms to some specific characteristics. The operational policy decision is constantly reviewed based on the latest resource utilization, PHBs and feedbacks from delivered traffic.

2.3 MPLS Networks

Multi-protocol Label Switching (MPLS) can be seen as an extension to the existing IP architecture [13]. It provides the equivalent capability of Virtual Circuit as in Asynchronous Transfer Mode (ATM) for IP packets [5]. In the traditional IP networks, forwarding decisions are made independently by routers based on the information in the IP headers. Hence, packets are transmitted to the destination in a connectionless fashion in which packets from the same source may be forwarded on different paths. This inevitably complicates packet controls and QoS provisions. On the contrary, MPLS provides connection-like packets transmission by making the forwarding decisions at the edge routers. As such, intermediate routers will have a better control of packet transmission and services provided.

The technique that MPLS [13] uses is known as label switching. When a packet arrives at the edge router on an MPLS domain, a short fixed-length 32-bit label is encoded into the packet header by the MPLS Label Edge Routers (LER) before dispatching it to the next hop. This label will be used by the intermediate router—Label Switch Router (LSR) to guide the packet forwarding process. Based on the label on an incoming packet, the LSR will perform label lookup to find the next hop and the corresponding outgoing label. Next, a label switching process is carried out in which the LSR will replace the existing label with the corresponding outgoing label such that the packet will be forwarded properly by the subsequent LSRs. This label switching process is conducted at every LSR that a packet transits until the last hop where the label will be removed. Hence, the path that a packet traverses through an MPLS domain is named as Label Switched Path (LSP). In order to satisfy the QoS requested, a LSP has to be set up in advance before using it for packet transmission.

The common signaling protocols used for LSP set up are RSVP-TE [14] and CR-LDP [15].

Fig. 2.5 illustrates the label switching process in an MPLS domain. For example, the LER *A* encodes a label 50 to the packet received from the source. When the LSR receives a packet with label 50, it performs label lookup and finds out that the corresponding outgoing label is 52. It then switches the label 50 with label 52 and forwards the packet to the next hop which is LER *B*. LER *B* removes the label and forwards the packet to the destination. The LSP used to forward the packet consists of LER *A*–LSR–LER *B*.

Fig. 2.5: Label Switching in MPLS.

Packet forwarding in MPLS domain is faster because label lookup is much easier as compared to prefix lookup in conventional IP forwarding [16]. With MPLS, packet forwarding can be done independent of the network protocols, and thus forwarding paradigms beyond the current destination-based technique can be easily supported. Therefore, it is possible to explicitly specify the path that a packet should traverse in order to meet certain service requirements or constraints. The explicit route mechanism in MPLS provides a critical capability that is currently lacking in the IP-based networks. MPLS also incorporates concepts and features from both Integrated

Services and Differentiated Services. E.g., MPLS allows bandwidth reservation to be specified over an LSP, and packets can be marked to indicate their priority.

The purpose of MPLS is not to replace IP routing but rather to enhance the services provided in IP-based networks. It allows the conventional hop-by-hop IP forwarding and LSP to coexist in order to bring the benefits of scalability and manageability. Non critical traffic may use IP forwarding to achieve scalability whereas critical traffic is able to use LSP to ensure that the QoS requested is satisfied. The flexibility of MPLS networks architecture has made it a suitable candidate for backbone networks [28] whereby it can be used to support Integrated Services and Differentiated Services.

2.4 MPLS Traffic Engineering

One of the major functionalities of MPLS networks is its support for explicit routing or Traffic Engineering (TE). It is often referred to the process of optimizing the performance of networks through efficient provisioning and better control of network flows [17]. Traffic engineering uses advanced route selection algorithms to provide resources to traffic trunks inside backbones and arrange traffic flows in such a way that maximizes the overall efficiency of the network. The common approach is to calculate traffic trunks based on flow distribution and then set up the traffic trunks as explicit routes with the MPLS protocol [18]. A traffic trunk is an aggregation of traffic flows of the same class that are placed inside a LSP. The combination of MPLS and traffic engineering provides IP-based networks with a set of advanced tools for service providers to manage the performance of their networks and provide more services at less cost.

Fig. 2.6: Traffic Engineering Processes.

As shown in Fig. 2.6, TE process can be divided into the following main parts [18],

- *Definition of control policies.* This process defines the service quality supported based on the users requirements. For example, the network may define different configurations from peak hours and off-peak hours.
- *Feedback mechanism.* This process acquires network performance-related information from the network. In the absence of actual data, the network can use synthetic workloads to obtain useful information.
- *Analysis mechanism.* This process analyzes the data obtained from the feedback mechanism. It can be either reactive or proactive. In reactive analysis, the network will try to identify the possible sub-optimal performance in the network. On the contrary, in proactive analysis, the network will try to anticipate the possible future performance.
- *Optimization mechanism.* Based on the analyses and a given objective, the network will optimize the performance accordingly. This process will make the actual configuration on the network. Subsequently, the optimized results will be reflected in the *policy control* process such that the network operators can inform end users of the services supported.

Fig. 2.7: LSP Routed on Not Over-utilized Path to Avoid Congestion.

For traffic engineering, the advanced route selection techniques that is often referred to as constraint-based routing, is used to calculate traffic trunks based on the optimization objectives. The objective may be to maximize the utilization of resources in the network or to minimize congestion in the network. For example in Fig. 2.7, traffic is routed on the path which is not over-utilized to avoid congestion. Typically, the optimal operation point is reached when traffic is evenly distributed across the network. With balanced traffic distribution, both queuing delay and loss rates are at their lowest points. To perform such optimization, the traffic engineering system often needs network-wide information on topology and traffic demands. Thus traffic engineering is typically confined to a single administrative domain. Nevertheless, an inter-domain TE that is designed for resolving conflicting routing issues across multiple networks is proposed in [29]. It makes use of Border Gateway Protocol (BGP) and network measurements to tune the TE parameters. Another tunable inter-domain routing scheme for TE is presented in [32]. In [30], the authors investigate the performance of TE using estimated traffic matrices. It is shown that TE algorithm that is able to optimize network performance based on exact traffic

matrices may not produce the best results. In fact, a combination of known traffic estimation and known TE techniques is able to give near optimum performance. Instead of using estimated traffic matrices, an online TE routing algorithm is proposed in [31].

In order to address both the traffic oriented and resource oriented performance objectives, LSPs are assigned with a number of attributes to facilitate traffic engineering. The important traffic trunk attributes are listed below [18]:

- Traffic parameter attribute.
- Policing attribute.
- Generic path selection and maintenance attribute.
- Priority attribute.
- Preemption attribute.

Traffic parameter and *policing attributes* are closed related in which they are used to monitor and control traffic. Traffic entering the network at the ingress node will be checked and noncompliant traffic will be marked accordingly or blocked. This will ensure incoming traffic will always adhere to the agreed contracts and prevent any greedy connections from starving the resources of the compliant connections. On the other hand, *generic path selection and management attribute* is concerned with the selection of the route taken by the LSP and the rules for the maintenance of paths that have been established.

Priority and *Preemption* attributes are used to characterize the relative importance of LSPs in a network. Priority attribute is essential for constraint based routing algorithm to route a LSP through a network. A high priority LSP is usually routed through relative favorable paths within a network as opposed with lower

priority LSP. Furthermore, priority attribute is the main parameter in preemption operation. Higher priority LSPs are able to preempt lower priority LSPs in the event of contention for network resources.

Two priorities are defined in which *set up priority* is used to determine whether a LSP can acquire the network resources from existing connections (e.g. trigger preemption) whereas *holding priority* is used to determine the relative importance of existing LSPs. MPLS networks can support eight priority levels with values ranging from 0 – 7, with 0 as the highest priority. Even though a LSP may be assigned different set up and holding priorities, it is required that the set up priority must be higher or equal to the holding priority [18]. This is to prevent the preemption of LSPs that have just been set up. Although MPLS-TE enables LSP to be set up on a relatively favorable path, it is possible that some of the links on this favorable path do not have sufficient idle bandwidth for the new LSP. In that case, the network may preempt a number of existing LSPs on the congested links.

The requirement of preemption in MPLS network was introduced in [17] but no specific preemption policy was proposed. The commercial routers normally implement the preemption policy based solely on the LSP priority level. This policy would result in high bandwidth wastage. The details of preemption and some of the existing strategies will be discussed later in this chapter.

2.5 Resource Allocation and Preemption

One of the fundamental problems in connection-oriented networks (e.g. MPLS, SONET/SDH and optical networks) is that of resource allocation. Before a source can send data to a destination, a connection has to be established with the desired

bandwidth and priority level. In the MPLS terminology, an LSP is equivalent to a connection. We use *connection* and *LSP* interchangeably to denote the traffic flow of the same class between a source–destination node pair. When a network is heavily loaded, a new connection may not be able to reserve the required bandwidth because one or more links are congested. In order to maximize the network’s resource utilization, the number of connections rejected due to link congestion should be minimized. For example, load balancing is used to prevent a part of the network from becoming over-utilized. Although connection rejection rate can be reduced by load balancing, network congestion may still happen if it is subjected to higher load conditions. In such a situation, a new connection may be blocked regardless of its priority level. This will compromise the integrity of priority levels and affect the QoS requested.

We observe that preemption is a practical problem which can arise from events such as oversubscription, node failure and link failure. However, without *a priori* knowledge of future connections, it would be difficult for the network to reserve sufficient resources for future mission-critical connections. If the new connection is relatively important, it would be inappropriate to deny access to it. For such a scenario, preemption is introduced as a mechanism to provide network resources to this important new connection by tearing down a number of existing lower priority connections at the congested links.

Before proceeding to describe the existing preemption schemes, it is worthwhile to investigate some of the strategies that can be used to minimize preemption in networks. These strategies include *Admission Control*, *Resource Reservation*, *Routing*, *Path Protection*, and *Preemption*. This will provide insights into the various

aspects of resource allocation in networks before introducing the importance of preemption in those strategies.

2.5.1 Admission Control

Admission control limits the load on networks by determining if an incoming request can be accommodated without disrupting the services of existing connections or traffic flows [5]. Using admission control mechanism, a connection will either be admitted or rejected. Admission control can be generally categorized into Parameter-based Admission Control (PBAC) and Measurement-based Admission Control (MBAC) [33]. In PBAC, the reservation of networks resources is done based on the information provided by the incoming requests. The typical parameters provided are such as average rate and peak rate. These parameters describe the upper bounds of the traffic generated by the requests. If the admission test succeeds, the new requests will be admitted. PBAC is simple because the networks do not need to predict the resources required by the new requests. On the contrary, MBAC makes admission decisions based on the network measurements. No explicit parameters are needed from the new requests. Therefore, MBAC is able to use the network capacity more fully than PBAC as no hard bound guaranteed is provided [33]. However, the higher network utilization is achieved at the expense of weakened service commitments.

In the traditional admission control approach, RSVP [2] is used to investigate the congestion level of network links and make resource reservation. This has limited scalability as the network has to keep per flow state and process per-flow reservation messages. In order to resolve this scalability issues, endpoint admission control [34] [35] are proposed to make admission decision based on the network performance

observed at the endpoints. This aggregate network performance will provide a clue as to whether admission can be granted to the new request. The endpoint admission control that incorporates Differentiated Services is presented in [36]. Admission control is not limited to wired networks, some of the admission control strategies for wireless networks are presented in [37-38].

Since no contention of network resources could happen as the new connections will be rejected when the network is highly loaded, preemption mechanism is practically not needed. However, this may violate the integrity of priority levels as new connections of high priority should be admitted and entitled to better networks services. In the events of node failure or link failure, preemption may be needed in order to route the high priority connections affected through alternative paths. Admission control alone is not able to provide the guaranteed services to connections; preemption remains a complementary mechanism to guarantee the resources needed by high priority connections. Although the admission control policy can be designed in such a way that low priority connections are rejected when the network anticipates high traffic rate in the near future, it may degrade network utilization if without *a priori* knowledge of future connections. Therefore, it is clear that admission control cannot be a substitute for preemption mechanism, instead, a combination of both will likely result in better network services.

2.5.2 Resource Reservation

Resource reservation mechanism is concerned with making reserves for incoming connections. The connections that require a minimum amount of bandwidth will inform the ingress routers of the resources needed. The request information will be

propagated throughout the whole network in order to find the feasible path. Once the path is identified, the routers will make the appropriate reservations for the connections [39–42]. Since the network services provided may differ between multiple network domains, most of the reservations are made on per-domain basis. To overcome this problem, the resource reservation strategy based on Bandwidth Broker (BB) is proposed in [43]. As shown in Fig. 2.8, this BB based strategy allows the BB from multiple domains to cooperate and coordinate the reservation process such that the end-to-end QoS requested is fully satisfied.

Fig. 2.8: Resource Reservation in Multiple Domains Using Bandwidth Broker.

Resource reservation can be divided into *Static* and *Active* [44]. In the static approach, the network resources assigned are fixed throughout the lifetime of the connections. This could lead to low utilization if the connections do not fully utilize the bandwidth allocated. On the contrary, active approach is more resource efficient as the unused bandwidth reserved will be reassigned temporarily to other connections. However, if the re-allocated bandwidth is needed by the original connection, it will be preempted from the existing connections that borrowed it. This indicates the requirement for a proper preemption technique to acquire resources from existing connections.

Most of the resource reservation strategies assume that requests are instantaneous, in which the connections only inform the ingress routers of the bandwidth needed marginally earlier than the actual arrival. Without advance knowledge of the incoming request, the network may reject the new connections if it is experiencing congestion. In order to ensure that sufficient bandwidth is reserved, book-ahead reservation [46–49] allows connections to make reservation in advance so that the network is able to reserve sufficient bandwidth when the connections arrive. This is able to minimize the conflicts of resources as the network may start rejecting some connections in anticipation of the connections that have made reservations. Due to the inherently higher pricing rates for advance reservations [50], only high priority connections are eligible to make the reservations, thus providing the means to for guaranteeing resources for high priority connections.

Although book-ahead reservation may minimize the need for preemption, it is not able to accommodate instantaneous high priority connections. Furthermore, book-ahead reservation may lead to low utilization as other connections are rejected before the arrival of the new connections. In [46], it is shown that book-ahead reservation can effectively reduce the number of connection preemptions but cannot entirely eliminate it. This motivates the requirement for preemption to resolve instantaneous connections resource conflicts and to achieve better utilization.

2.5.3 Routing

Routing is used to search for the feasible path that a connection can use to reach its destination. The most common routing objective is to search for the shortest path, which is the path with the minimum hop count. This single routing metric, i.e. hop

count, can be replaced by other metrics, e.g. delay, depending on the connections requirements. In general, routing algorithms such as *Dijkstra's* algorithm and *Bellman-Ford* algorithm are able to provide the path with this single metric. However, recent advancements in QoS provisioning drive the need for *multi-constraints QoS routing* [51–53]. The path is no longer bounded by a single metric, but multiple metrics such as hop count, delay, bandwidth, and reliability.

The routing metrics can be divided into three classes. Let $d(i,j)$ be a metric for the link (i,j) . For any path $P = (i, j, k, \dots, m)$, the overall metric d is

- *Additive* if $d(P) = d(i,j) + d(j,k) + \dots + d(l,m)$
- *Multiplicative* if $d(P) = d(i,j) * d(j,k) * \dots * d(l,m)$
- *Concave* if $d(P) = \min\{d(i,j), d(j,k), \dots, d(l,m)\}$

According to this definition, metrics such as delay, jitter, cost and hop-count are additive, reliability is multiplicative and bandwidth is concave. A well-known theorem in routing problem is that the path subject to constraints of two or more additive and/or multiplicative metrics is NP-complete [54]. The computationally feasible combinations of metrics are bandwidth and one of the other metrics.

One of the areas of routing that is pertinent to preemption issues is minimum interference routing such as MIRA [55], LIOA[56], [57-58]. These algorithms focus on minimizing blocking ratio by reducing future interference on critical network links. The critical network link is defined as the link that is highly probable to be used by future connections. By routing the existing connections on the links that exclude the critical links as much as possible, future connections are likely to be routed successfully. This in turn implies that minimum interference routing can simultaneously minimize preemption probability as more connections can be

accommodated by the network. Although lower blocking probability is observed in these routing strategies, high priority connections may still be rejected due to network congestion.

In order to minimize preemption events, preemption-aware routing [59] is proposed to search for the path that takes into consideration of routing metrics and preemption objective. Two preemption objectives are defined—bandwidth based and priority based. In bandwidth based preemption, the main objective is to minimize the total amount of bandwidth preempted. On the other hand, priority based preemption aims at minimizing the priority levels preempted. With this preemption-related information, the routing algorithm is able to select the path that minimizes the total number of preemptions. The main motivation for minimizing preemption is to reduce connection re-routing such that network stability is not adversely affected. Similarly, F. Blanchy *et al.* [60] proposed a score function such that the path chosen triggers the least number of preemptions. The preemption objective used in [60] is bandwidth based. The preemption aware routing in [61] is designed by taking into account both the instantaneous connections and book-ahead connections. Given an incoming instantaneous connection, the existing connections and book-ahead connections, the algorithm is able to compute the preemption probability. This information is used to guide the routing algorithm to search for the feasible path that minimizes preemption.

Note that routing may be a useful mechanism for minimizing preemption events in networks. However, preemption mechanisms are still needed to decide which combination of connections to be preempted. In fact, both routing and preemption can play complementary roles to enhance the network performance. An effective preemption algorithm is able to tear down the existing connections such that the

resulting disruption is minimized. The routing algorithm is responsible for selecting the path that minimizes the total number of connections preempted.

2.5.4 Path Protection

A possible alternative to avoid triggering preemption is through the implementation of path protection. Generally, MPLS with LSP protection [62–66] is a mechanism to protect LSP in the event of failures. A disjoint backup path is routed side-by-side with the primary path and it is used to carry the traffic if failure occurs on the primary path. Two methods of providing LSP protection exist, i.e. *global* re-route and *local* re-route as shown in Fig. 2.9 and Fig. 2.10. In global re-route, the alternative path is established between the source and the destination. On the other hand, local re-route constructs the alternative path between the two end nodes of the failed link.

Fig. 2.9: Global Re-route.

Fig. 2.10: Local Re-route.

Global re-route is more resource efficient as the backup path is setup between the source and destination. The path chosen could be of the same length (hop counts)

as the primary path. However, the overhead involved has a higher complexity as the network needs to tear down the resources occupied on all the links of the primary path. It is shown in [67–69] that local re-route for LSP protection can significantly reduce the control overhead involved and expedite the re-routing process. By re-routing the path between the two adjacent nodes of a failed link, the network reduces the time and effort to tear down the primary path and setup the backup path. However, local re-route is less resource efficient as the resulting backup path will be longer.

It is possible to implement path protection for low priority connections that are subject to preemption. The backup path routed side-by-side with the primary path can be used to carry the traffic if the connection is preempted. This will ensure that the ongoing services are not interrupted. Although path protection mechanism can be used in preemption problem, it is an expensive operation because excessive resources are required to establish the backup paths. The resources used to route the backup paths cannot be shared by other connections. Therefore, it remains an operation reserved primarily for high priority LSPs which carry critical data.

2.5.5 Preemption

Connection preemption is defined as the process of tearing down one or more existing low priority connections in order to provide sufficient resources to the new high priority request. Preemption normally stems from *oversubscription*, *link failure* or *node failure*. The general condition for triggering preemption is resource scarcity, in which the network is not able to support the resources requested by all the connections.

An example of preemption is illustrated in Fig. 2.11. All the links in the simple network topology are assumed to be able to support only one connection at any one time. Initially, low priority connection 1 is established between the nodes $B-C-A$. The new high priority connection 2 between the nodes $D-C-A$ causes resource conflicts on link $C-A$. In order to provide resources for connection 2, low priority connection 1 is preempted. The network will teardown all the resources occupied by connection 1. Finally, connection 2 is setup on the path $D-C-A$.

Fig. 2.11: Preemption Processes.

2.5.5.1 Preemption in ATM

Various studies had been done on preemption policies in the literature. Garay *et al.* [70] introduced the call preemption techniques for Asynchronous Transfer Mode (ATM) networks. However their algorithms are meant for the centralized network

environment. In their paper, Garay *et al.* proved that the problem of selecting which connections to preempt in order to minimize the number of connections to be preempted or to minimize the amount of bandwidth to be preempted is NP-complete [71]. Since the problems are computational intractable, heuristics algorithms are proposed. The heuristics algorithms have the computational complexity of the order of $O(n \cdot m^2)$ where n is the total number of hops along the preempting path, and m is the size of the set which contains all existing connections that have at least one link in common with the path selected for preemption. The connections included in set m must have priority levels lower than the preempting connection. The algorithms try to minimize the total number of connections preempted by preempting the connection that has the highest number links in common with the selected path.

Another preemption policy for ATM network was introduced by M. Peyravian *et al.* in [72]. As opposed to [70], the authors [72] proposed a decentralized approach in which each link manager acts independently of each other to determine the connections for preemption. Two optimal decentralized connection preemption algorithms are developed to minimize the disruption to existing connections and satisfy the constraints of higher priority connections. The algorithms are executed locally by each link along the chosen path if not enough bandwidth cannot be allocated on the link. Three preemption criteria are proposed to evaluate the existing connections for preemption,

1. *Bandwidth.* By minimizing the total amount of bandwidth preempted on a link, more connection can be accommodated. This criterion is designed for the objective of resource efficiency.
2. *Priority.* In order to preserve the integrity of priority levels, a connection can

only preempt existing connections of lower priority. By preempting from the lowest priority, the network will be able to allocate more resources for high priority connections.

3. *Number of Connections*. By minimizing the number of connections preempted, the network reduces service interruption to end users.

Different ordering of the preemption criteria are proposed in [72]. Performance comparisons between the decentralized and centralized algorithms are presented in [73]. The results show that no significant performance is observed in the centralized algorithms. In fact, both the algorithms perform similarly. However, the decentralized algorithms have better advantage due to its simplicity in real implementation.

The preemption policy for military ATM network is presented in [74]. The algorithm is designed to perform connection precedence and preemption in ATM networks with a central controller, as found in a Demand Assigned Multiple Access (DAMA) satellite network or wireless ATM (WATM) network. The main purpose of the algorithm is to prevent lockout by considering connection precedence when making the admission decision and preempting lower precedence connections when the requested resources are not available. The author also presented the solutions for implementation issues such as backward compatibility, security, and restoration of preempted connections in case of preemption failure. Another version of the algorithm is introduced in [75], which uses ATM Forum standard signaling to provide the preemption feature in backbone ATM networks.

The preemption strategy that is based on connection service time is proposed in [76]. The objective of the preemption strategy is to maximize the throughput accrued. The authors proved that a single parameter – the duration of a connection, is

sufficient to design an algorithm to enhance the throughput. By assuming that connection service time is known in advance, the algorithms are able to preempt the connections that are far from completion. These strategies are best suited for networks where connection service time is declared before being admitted. In [77], strategies were developed to improve the performance of the network by reducing the overhead due to preemptions. Assuming service times to be known in advance, this improvement is achieved by allowing preemption only when the remaining service time is greater than the overhead.

2.5.5.2 Preemption in Wireless Networks

The preemption policies for wireless mobile networks are introduced in [78]. The authors presented an analytical model for integrated real-time and non-real-time service in a wireless mobile network with priority reservation and preemptive priority handoff schemes. The preemption schemes are aimed to provide better QoS for third generation wireless mobile networks that is characterized by more provisions for real-time traffic. The authors divided the service calls into four different types, which are real-time originating calls, non-real-time calls, real-time handoff calls, and non-real-time handoff calls. Since handoff calls are relatively more important than originating calls based on the sentiment that end users will be more dissatisfied if ongoing calls are terminated abruptly, real-time handoff calls have higher priority than real-time originating calls as well as the non-real-time calls. Preemption is triggered based on these different priority levels. In [79], two handoff schemes are proposed—preemptive scheme and non-preemptive scheme. In the non-preemptive scheme, only real-time calls are allowed to gain access to the idle non real-time channels. However, in the preemptive scheme, non real-time calls can borrow the real-time channels. The

preemption mechanism allows real-time calls to preempt non real-time calls. The analytical results show that the preemptive scheme achieves better utilization.

2.5.5.3 Preemption in MPLS

J.C. de Oliveira *et al.* [80-81] presented a new preemption policy complemented with adaptive scheme for MPLS networks to minimize rerouting. The optimization criteria used are similar to [72], i.e. the amount of bandwidth to be preempted; priority of LSPs to be preempted; number LSPs to be preempted. The preemption policy also uses decentralized approach which is relatively easier to be integrated to the current Internet protocol as the parameters required by the policy are readily available from routing protocols such as Open Shortest Path First (OSPF). Instead of ordering the preemption criteria and optimizing them sequentially, an objective function is proposed to evaluate the criteria collectively. The coefficients attached to the different criteria enable service provider to flexibly adjust the function to reflect different optimization objective desired. Instead of preempting the LSPs directly, the adaptive scheme allows existing LSPs to reduce the bandwidth occupied such that the new LSP is able to route successfully. The algorithm allows the existing connection to reduce up to 50% of its bandwidth. These preemption policies are evaluated based on an MPLS test-bed as presented in [82]. In [83], performance comparisons between randomly selected and pre-determined preemption criteria are conducted. It is shown that preemption based on random selection is able to perform comparably to existing preemption strategies [72],[81]. However, the network performance beyond the simple network topology used remains to be verified.

An optimal preemption algorithm is proposed in [84] with the same three preemption criteria as in [72]. The algorithm will first preempt existing connections

from the lowest priority. A combination of connections based on the rest of the criteria is selected for preemption at the priority level that exceeds the bandwidth requirement. This preemption algorithm has exponential complexity. T. Shan *et al* [85] present a bandwidth allocation and preemption strategy for Differentiated-Service-Aware networks. The bandwidth is divided between high priority and low priority applications. Under oversubscription condition, idle bandwidth assigned for high priority applications can be temporarily used by low priority connections. Preemption is used to acquire the shared bandwidth if needed. In [86], a backward connection preemption algorithm is introduced. It utilizes MPLS-TE framework to collate information about the existing LSPs and network links. A preemption decision can be then made so as to minimize the number of LSPs preempted. Heuristic bandwidth allocation with preemption proposed in [87] uses three different utility functions (step, linear and concave) to represent the requests. The marginal worth for each request is derived so that for each preemption event, the connections with the lowest marginal worth are the first to be preempted.

Matthew R. Meyer *et al.* [88] proposed *soft preemption* scheme in which the to-be-preempted LSPs are re-routed before being torn down. This draft proposed the use of additional signaling and accounting mechanisms to alert the ingress Label Edge Router (LER) of the preemption that is pending and allow for temporary over-provisioning while the preempted LSP is re-routed in a non-disruptive fashion (make-before-break) by the ingress LER. During the period that the tunnel is being re-routed, link capacity is over-provisioned on links where soft preemption has occurred. This draft proposed a suite a protocol to be used in soft preemption but did

not make any insight study into path selection and re-routing schemes. Table 2.1 provides a brief description of the existing preemption schemes discussed above.

Table 2.1: A List of Existing Preemption Schemes.

No.	Authors	Application	Description
1.	J. Garay <i>et al.</i> [70]	ATM	Propose a centralized algorithm that minimizes the number of connections and/or bandwidth preempted.
2.	M. Peyravian <i>et al.</i> [72]	ATM	Propose decentralized algorithms. Utilize three preemption criteria, i.e. bandwidth, priority and number of connections.
3.	S. Poretsky [74]	ATM, Wireless ATM	Prevent lockout by preempting lower precedence connections. Present solutions for backward compatibility, security and restoration.
4.	A. Bar-Noy <i>et al.</i> [76]	ATM	Propose centralized algorithms that maximize throughput by considering connection service time.
5.	J. Wang <i>et al.</i> [78]	Wireless mobile networks	Provide better QoS by giving priority to handoff calls and real-time calls.
6.	W. Li <i>et al.</i> [79]	Wireless mobile networks	Allow non real-time calls to borrow real-time channels. Real-time calls can preempt non real-time calls.
7.	J.C. de Oliveira <i>et al.</i> [81]	MPLS	Propose a decentralized algorithm that combines three different criteria, i.e. bandwidth, priority and number of connections.
8.	V. Stanisic <i>et al.</i> [83]	MPLS	Conduct performance comparisons between randomly selected preemption criteria and pre-determined preemption criteria.
9.	T. Shan <i>et al.</i> [85]	MPLS	Allow low priority connections to use idle bandwidth assigned to high priority connections. High priority connections can preempt low priority connections.
10.	L. Lei <i>et al.</i> [86]	MPLS	Use MPLS-TE capability to minimize the number of connections preempted.
11.	P. Dharwadkar <i>et al.</i> [87]	MPLS	Derive connection marginal worth based on three different utility functions. Preempt connections with the lowest marginal worth.
12.	M.R. Meyer <i>et al.</i> [88]	MPLS	Re-route connection before it is preempted.

2.6 Summary

From the discussions in the above sections, it is worthwhile to note that preemption is a practical problem that could arise from oversubscription, node failure or link failure. Although techniques such as admission control, resource reservation, routing,

and path protection can minimize preemption events, none of them can totally eliminate it. A well designed preemption strategy is needed to ensure that high priority connections can gain access to network resources in the event of competition such that it could complement those techniques discussed in the previous sections. Furthermore, the network needs to determine a proper combination of existing connections for preemption. These connections must be selected based on a well designed methodology such that the preemption objective is optimized, e.g. bandwidth minimization. One area in preemption that is constantly overlooked by current preemption strategies is the service interruption. As preempted connections will be terminated, the ongoing services will be interrupted as well. This would inevitably affect end users' perception on the network services provided.

In this thesis, the author proposes that preemption with re-routing (*soft preemption* [88]) is a better strategy as to-be-preempted connections are re-routed to avoid service interruption. This strategy will essentially reorganize the network connections such that more connections can be admitted without excessively disrupting existing services. As such, the network will achieve higher throughput. The network performance of routing with preemption is investigated using simulation. In addition, instead of using routing technique to minimize preemption, the author proposes that routing algorithm can make use of preemption for the path searching process. This shifts preemption from the traditional passive domain into the active domain, meaning that the routing algorithm can trigger preemption more actively in order to route a particular connection through the more favorable path. Finally, the author also seeks to model the network preemption mechanism through

Stochastic Petri Nets (SPN). This model is able serve as a common platform for performance comparisons for different preemption schemes.

3. Preemption with Re-routing to Minimize Service Disruption

Resource allocation is a fundamental problem in connection-oriented networks, (e.g. MPLS, SONET/SDH and optical networks). Before a source can send data to a destination, a connection has to be established with the desired bandwidth and priority level. However, without *a priori* knowledge of future connections, it would be difficult for the network to reserve sufficient resources for the future mission-critical connections. If the new connection is relatively important, it would be inappropriate to deny access to it. For such scenarios, preemption is introduced as a mechanism to provide network resources to this important new connection by tearing down a number of existing connections of lower priority at the congested links. For example, preemption is implemented in MPLS networks [80–87] and ATM networks [70], [72–76] by using the criteria such as connection's priority level and bandwidth.

However, a common problem faced by the existing preemption schemes is that *service disruption* is incurred when existing connections are terminated. This results from the interruption of service experienced by end-users when ongoing transactions are stopped and the services conducted by the preempted connection is lost. This will adversely affect the network throughput and it is especially detrimental when a number of ongoing connections are preempted in order to admit one incoming

connection. It may be noted that although preemption will cause service disruption, the justification to trigger preemptions may be that some mission critical connections may have to be admitted or that opportunities may arise where higher revenues may be obtained by admitting new incoming connections in spite of the disruptions.

To overcome service disruption, Meyer *et al.* [88] proposed a *soft preemption* scheme to re-route the connections before termination. As existing connections are re-routed and switched to alternative paths, end users will experience bursts of delays but will be otherwise unaware of the underlying preemption process. The work completed will not be affected if soft preemption is successfully triggered for an existing connection. This will minimize the damage caused by preemptions and help to alleviate the loss of useful network utilization. However, the soft preemption scheme [88] only proposes a suite of protocol to trigger the re-routing process. The authors suggest that preemption algorithms such as those introduced in [81] could be used to determine the combination of existing connections for preemption.

The main motivation of this chapter is to minimize the service disruptions caused by preemptions. The general objective here is to satisfy the requests (e.g. bandwidth) of higher priority connections as far as possible, even with preemptions, while simultaneously reducing the detrimental effect this may have on the service provided to lower priority connections. To achieve this, two distinct algorithms are proposed in this chapter, namely a centralized algorithm and a decentralized algorithm. Both algorithms employ soft preemption as the fundamental mechanism to minimize service disruption. The centralized algorithm greedily searches the network for re-routable connections (i.e. connections which can be re-routed in the existing topology and usage conditions) based on network-wide information. Once

preemption is triggered, these re-routable connections are preempted ahead of non re-routable connections so that the number of connections which have to be terminated by force (i.e. for preemption) is minimal. Simulation results show that the centralized algorithm greatly reduces service disruption as compared to existing preemption schemes. For feasible deployment, the author also proposes a decentralized preemption strategy that only uses local information. The decentralized algorithm assigns costs to the network links such that highly loaded links are associated with higher costs. Thus, by routing the new connections using the minimum cost path, fewer existing connections would need to be preempted as the links selected are less congested. The decentralized algorithm reduces service disruption by first preempting existing connections that can be locally re-routed, before preempting other connections. The simulation results show that the network performance achieved by this decentralized approach is comparable to that of the centralized strategy.

3.1 Problem Formulation

In [17], the requirement for *priority* and *preemption* parameters is highlighted in order to address both the traffic-oriented and resource-oriented performance objectives for Traffic Engineering in a MPLS network. The preemption attribute is used to determine whether a new Label Switched Path (LSP) can preempt an existing LSP. Two different types of priority are proposed where a LSP with high *setup priority* can preempt an existing connection with low *holding priority*. Although the preemption algorithms formulated are based on the architecture of a MPLS network, the algorithms will also be applicable, in general, to other types of connection-

oriented networks. No distinction is made in this work between the *setup priority* and *holding priority* since both the priorities are assumed to have the same value.

Each network connection is represented by $c = (b, p)$, where b denotes bandwidth of the connection and p denotes its priority level. The priority level p can range from 0 to 7, with 0 represents the highest priority. Similarly, a new connection arriving at the network is denoted by $c_{new} = (b_{new}, p_{new})$ where both the parameters b_{new} and p_{new} are used to determine the combination of connections to be preempted (if any) to satisfy the new bandwidth request. Let the network be represented by a directed graph $G = (N, E)$, where N is the set of nodes and E is the set of edges. It is assumed that each link $E_{ij} \in E$ is associated with a parameter U_{ij} to denote the total idle bandwidth available on the link between nodes i and j . When a new connection occupies the resource on the link, this parameter is updated by the following equation, $U_{ij} = U_{ij} - b_{new}$. Whenever an existing connection with bandwidth b_{old} on the link completes its service, this parameter is updated by $U_{ij} = U_{ij} + b_{old}$.

Suppose a new connection c_{new} arrives and finds that the network is not directly able to find sufficient network resources for it (i.e. no path can be found from source to destination with adequate free bandwidth on its entire links to satisfy the bandwidth required by this connection). Let $l = \{s, 1, 2, \dots, t\}$ be the path identified for the new connection which contains one congested link, say $E_{ij} \in l$. Given that there are R numbers of existing connections occupying the network resources on link E_{ij} , we can define the set of existing connections, bandwidth and priority levels respectively with the following notations.

$$\mathbf{C}_R^{ij} = \{c_1^{ij}, c_2^{ij}, \dots, c_R^{ij}\}. \quad (3-1)$$

$$\mathbf{B}_R^{ij} = \{b_1^{ij}, b_2^{ij}, \dots, b_R^{ij}\}. \quad (3-2)$$

$$\mathbf{P}_R^{ij} = \{p_1^{ij}, p_2^{ij}, \dots, p_R^{ij}\}. \quad (3-3)$$

Note that the network is not able to route the new connection on the link E_{ij} , if the idle bandwidth available on the link is lower than the bandwidth requested, i.e. $U_{ij} < b_{new}$. However, the new connection can still be accommodated by preempting some of the existing connections with lower priority levels. Let T_{ij} denote the sum of existing bandwidth that can be preempted from the existing connections on link E_{ij} . Then, the standard explicit expression for T_{ij} is given as follows,

$$T_{ij} = \sum_{r=1}^R b_r^{ij} I_{\xi}(b_r^{ij}), \quad (3-4)$$

$$I_{\xi}(b_r^{ij}) = \begin{cases} 1 & \text{if } b_r^{ij} \in \xi, \\ 0 & \text{if } b_r^{ij} \notin \xi. \end{cases}$$

$$\xi = \{c_r^{ij} : p_{new} < p_r^{ij}; c_r^{ij} \in \mathbf{C}_R^{ij}\},$$

where $I_{\xi}(b_r^{ij})$ is the indicator function for the event ξ that the new connection has higher priority than the existing connections on link E_{ij} . Therefore, preemption can only take place when $U_{ij} + T_{ij} \geq b_{new}$. To summarize, one can specify three possible scenarios on link E_{ij} when a new connection arrives. These are described as follows,

Scenario 1: No preemption will be triggered if idle bandwidth $U_{ij} \geq b_{new}$.

Scenario 2: Preemption will be triggered if the bandwidth requested exceeds idle bandwidth but is less than the total bandwidth available through preemption i.e.,

$$U_{ij} < b_{new} \leq U_{ij} + T_{ij}.$$

Scenario 3: c_{new} will be rejected if $b_{new} > U_{ij} + T_{ij}$ or if the requested bandwidth exceeds the link capacity.

For a connection that will not trigger preemption, all the links $E_{ij} \in l$ must satisfy scenario 1. Otherwise, preemption can be triggered to route the new connection if scenario 2 is satisfied. The above formulation is merely used to investigate whether a connection will trigger the preemption mechanism or not. In the event that $b_{new} = U_{ij} + T_{ij}$, all the existing connections on link E_{ij} with lower priority will be preempted to admit the new connection. This is because the sum of bandwidth that can be preempted, T_{ij} , equates exactly the shortfall of idle bandwidth needed by the new connection. However, if $b_{new} < U_{ij} + T_{ij}$, we would like to determine the combination of existing connections to be preempted to minimize service disruption.

3.2 Preemption Framework

In this section, the author seeks to formulate a preemption solution framework using a single-link approach. This framework is used as the basis for developing the large scale strategies presented later in the subsequent sections. Recall that the combination of connections to be preempted can be decided based on three preemption criteria [72] – (i) bandwidth, (ii) priority level and (iii) number of connections preempted. This section adopts the same preemption criteria as in [72] and also explores the preemption framework in [81]. However, in order to preserve the integrity of connection priority level, the author proposes that connections with the lowest priority level are preempted before the connections of the next (higher) priority level.

As similar to [81], for the link $E_{ij} \in l$ that triggers preemption, the set $\mathbf{M}_K^{ij} \subseteq \mathbf{C}_R^{ij}$ is denoted as the set of existing connections that can be preempted. Integer K is the cardinality and the set can be obtained by the following routine [81] in Fig. 3.1.

```

For  $r = 1$  to  $R$ 
    if  $p_{new} < p_r^{ij}$ 
         $\mathbf{M}^{ij} := \mathbf{M}^{ij} \cup \{c_r^{ij}\}$ 
    End if
End for

```

Fig. 3.1: The Routine to Obtain the Set of Preemptable Connections.

Various authors have proposed to perform preemption by using a specifically pre-determined ordering pattern of the preemption criteria, as in [72] and [84]. However, extensive investigation may be needed in order to find out if a particular ordering of the preemption criteria will result in the desired performance. To attain a flexible preemption scheme with respect to the three different criteria, [81] proposed that the following objective function (3-5) can be used to evaluate the combination of connections to be preempted, The primary motivation for (3-5) is to minimize the number of actual connections preempted.

$$\text{Min } F(\mathbf{z}) = \alpha(\mathbf{z} \cdot \mathbf{y}^T) + \beta(\mathbf{z} \cdot \mathbf{1}^T) + \gamma(\mathbf{z} \cdot \mathbf{b}^T), \quad (3-5)$$

where for all $c_k^{ij} \in \mathbf{M}_K^{ij}$,

$$\mathbf{z} = \{z_1, z_2, \dots, z_K\}, \quad z_k \in \mathbf{z}, \quad z_k = \begin{cases} 1 & \text{if } c_k^{ij} \text{ is preempted,} \\ 0 & \text{otherwise.} \end{cases} \quad (3-6)$$

$$\mathbf{y} = \{y_1, y_2, \dots, y_K\}, \quad y_k \in \mathbf{y}, \quad y_k = P_{\max} - p_k^{ij}, \quad (3-7)$$

$$\mathbf{b} = \{b_1, b_2, \dots, b_K\}, \quad b_k \in \mathbf{b}, \quad b_k = b_k^{ij}, \quad (3-8)$$

$$\mathbf{1} = \{1, 1, \dots, 1\}, \text{ is a unit vector with dimension } K. \quad (3-9)$$

In equation (3-5), α , β , and γ are scalars that can be tuned to reflect the preemption objective to be achieved. The variable $P_{\max}$ in (3-7) is the maximum number of priority levels supported. For example in MPLS networks [13], if the priority levels supported range from (0 – 7), then $P_{\max} = 8$. On the right hand side of (3-5), the first term expresses the cost of priority level preempted, followed by the number of connections preempted as represented in the second term, and the sum of bandwidth preempted in the third term.

Let the bandwidth amount that needs to be preempted on link E_{ij} as $V_{ij} = b_{new} - U_{ij}$, [81] minimizes (3-5) to obtain the set of connections for preemption as follows,

Single Link Optimal Preemption (SOP)

Given $\mathbf{y}, \mathbf{b}, \alpha, \beta, \gamma, V$

Find $\mathbf{z}$ by:

$$\text{Min } \alpha(\mathbf{z} \cdot \mathbf{y}^T) + \beta(\mathbf{z} \cdot \mathbf{1}^T) + \gamma(\mathbf{z} \cdot \mathbf{b}^T) \quad (3-10)$$

Subject to:

$$\mathbf{z} \cdot \mathbf{b}^T \geq V \quad (3-11)$$

The coefficients α , β , and γ can set to 0 or 1 respectively, in order to single out the effects of the different preemption criteria. For example, by setting $\alpha=1$, $\beta=0$, and $\gamma=0$, the preemption scheme minimizes bandwidth only as similar to [70]. However, in order to ensure that every criterion exerts relatively the same impact, the coefficients can be normalized in such as way that $\alpha = 1/P_{\max}$, $\beta = 1$, and $\gamma = 1/B_{\max}$ where $B_{\max}$ is the maximum amount of bandwidth than can be requested.

One of the essential problems facing the existing scheme [81] is that it may preempt a connection of higher priority before other lower priority connections if it is evaluated to have the lowest cost in $F(\mathbf{z})$. This is possible if the higher priority connection occupies the exact amount of bandwidth that needs to be preempted. However, this preemption of higher priority connection is inconsistent with the integrity of priority levels and may jeopardize users' perception on the QoS provided. Thus, it is more appropriate to preempt existing connections from the lowest priority level. A new solution is needed to overcome this problem. Let S_p be the sum of bandwidth from existing connections with priority level p ,

$$S_p = \sum_{k=1}^K b_k^{ij} I_\phi(b_k^{ij}), \quad (3-12)$$

$$I_\phi(b_k^{ij}) = \begin{cases} 1 & \text{if } b_k^{ij} \in \phi, \\ 0 & \text{if } b_k^{ij} \notin \phi. \end{cases} \quad (3-13)$$

$$\phi = \{c_k^{ij} : p_k^{ij} = p; c_k^{ij} \in \mathbf{M}_K^{ij}\}, \quad (3-14)$$

where $I_\phi(b_k^{ij})$ is an indicator function for the event ϕ that the existing connection in $\mathbf{M}_K^{ij}$ has the same priority as p . The preemption scheme that is based on the newly proposed priority first solution (OP-Prio) is described in Fig. 3.2. This new OP-Prio algorithm essentially preempts connections from the lowest priority until the condition when $S_p > V$, at which it will use the objective function (3-5) to select the connections. In order to truly preserve the integrity of priority level, only connections with priority p is included in the set $\mathbf{M}^{ij}$.

```

Given  $U_{ij} + T_{ij} \geq b_{new}, V$ 
For  $P = (P_{max} - 1)$  to  $(p_{new} + 1)$ 
  If  $V > 0$ 
    If  $S_p \leq V$ 
      Preempt all connections with priority  $p$ 
       $V = V - S_p$ 
    Else
       $\mathbf{M}^{ij} := \mathbf{M}^{ij} \cup \{c_r^{ij} : p_r^{ij} = p\}$ 
      Preempt connections using SOP
    End if
  End for

```

Fig. 3.2: OP-Prio Preemption Routine.

Table 3.1: Bandwidth and Priority Information of 16 Connections on a Single Link.

Connection	C1	C2	C3	C4	C5	C6	C7	C8
Bandwidth	20	10	60	25	20	1	75	45
Priority	1	2	3	4	5	6	7	5
Connection	C9	C10	C11	C12	C13	C14	C15	C16
Bandwidth	100	5	40	85	50	20	70	25
Priority	3	6	4	5	2	3	4	7

Fig. 3.3 illustrates the performance comparison of these different preemption strategies based on the connection information in Table 3.1. For this, SOP is solved by using CPLEX. The resource of the single link is assumed to be fully occupied by the 16 connections in Table 3.1. The bandwidth of the new connection is varied from 1–320 units. Different combinations of connections may be preempted based on the preemption strategies. As the preemption cost reflects the bandwidth and the number of connections preempted, lower preemption cost indicates a better preemption strategy. Since OP-Prio exhibits relatively good performance as compared to the other algorithms, the author believes that it is feasible to apply OP-Prio with the advantage of preserving the connections' priority levels.

Fig. 3.3: Performance Comparisons of Different Preemption Strategies.

3.3 Heuristics Preemption Algorithm

The optimization of SOP is a zero-one knapsack problem [90] which is NP-complete. Exhaustive search is required to produce the global optimum. A heuristic algorithm with comparable performance is needed to make the preemption decisions in shorter time. The heuristic algorithm proposed in [81] involves the computation of a preemption index for every existing connection in $\mathbf{M}_K^{ij}$. The value of the preemption index for the $c_k^{ij} \in \mathbf{M}_K^{ij}$ connection is given by the following equation.

$$F_h(c_k^{ij}) = \alpha y_k + \beta \left[\frac{1}{b_k} \right] + \gamma [b_k - V]^2. \quad (3-15)$$

On the right hand side of (3-15), the first term is used to denote the priority level; the second term represents the number of connections preempted and the third term minimizes the bandwidth preempted. Preemption is done with the increasing order of the preemption index. From the results given in [81], it is noticeable that the heuristics algorithm suffers from the over-provisioning of bandwidth which is caused by the failure to update the bandwidth that needs to be preempted. Consequently, the heuristics algorithm results in a much higher preemption cost as compared. The author seeks to amend this bandwidth over-provisioning problem by introducing an additional step as shown in Fig. 3.4.

Fig. 3.4: Modified Heuristics Algorithm (MHA).

In order to minimize the over-provisioning of bandwidth, the author proposes that the algorithm updates the amount of bandwidth that needs to be preempted such that $V = V - b_k^{ij}$ where b_k^{ij} is the bandwidth of an existing connection preempted. The flow diagram of the newly modified heuristics algorithm shows that when a network link triggers preemption, it will firstly assign an index value calculated based on (3-15) to every existing connection. Once the calculation is done, the link will preempt

the connection with the lower index value and subsequently the amount of bandwidth required is deducted by the latest preempted bandwidth. This process continues until the bandwidth required by the new connection is satisfied.

As similar to the objective function (3-5) used in SOP, the preemption index (3-15) in the heuristics algorithm may lead to the preemption of higher priority connections before lower priority ones. To overcome this, the author implements the same strategy as shown in Fig. 3.2 to preempt the connections from the lowest priority. For the condition when $S_p > V$, the modified heuristics preemption algorithm in Fig. 3.4 will be triggered to select the right combination of connections. Since the connections are preempted from the lowest priority, the first term on the right hand side of (3-15), αy_k , is not needed and is thus eliminated in this study. This modified heuristics approach is referred to as *MHA*.

Using the same information as given in Table 3.1, Fig. 3.5 presents the performance of MHA, and that of the heuristics in [81] as compared to that of the OP-Prio approach. The preemption cost is calculated by using the objective function (3-5). From our investigations, MHA achieves the best performance when $\beta = \gamma = 1$ (Recall that αy_k is not needed in MHA as the algorithm preempts from the lowest priority, the term is dropped for the rest of the chapter). In Fig. 3.5, ‘Heuristics’ [81] gives the highest cost because it is not able to minimize the sum of bandwidth preempted effectively, i.e. it lacks the extra new step in Fig. 3.4 to update the latest level of bandwidth that needs to be preempted.

Fig. 3.5: Performances of OP-Prio, Heuristics and MHA, ($\alpha=\beta=\gamma=1$).

MHA is able to match closely the preemption cost of OP-Prio. By optimizing the last term in (3-15), $\gamma[b_k - V]^2$, MHA minimizes bandwidth preempted and thus the number of connections preempted. The second term in (3-15), $\beta[1/b_k]$, is able to further minimize the number of connections preempted when two or more connections are evaluated to have the same value on the last term. As an example, if there are two connections such that $V = 10$, $b_1 = 8$ and $b_2 = 12$, the last term for both connections will have the same value. However, $F_h(c_2)$ is lower due to the second term and thus only one connection is preempted in this example.

To better understand the performance of these different strategies, extensive simulations written in C++ were conducted on a single link using Poisson arrival process with different rates and an exponentially distributed service time with mean

of 500s. The link has a capacity of 500 bandwidth units and the connections can request uniformly distributed bandwidth $U(10 - 50)$. The priority level is randomly selected from the integer (0 - 7). When a connection arrives that cannot find sufficient idle bandwidth on the link, then it will either preempt a number of existing connections (if the link has sufficient preemptable bandwidth) or it will be rejected.

The primary performance metric is the link throughput. For a particular connection, the throughput accrued is defined as the dot product of bandwidth assigned and the total service time. It is also defined that the total throughput is the sum of throughput from the connections that have completed successfully. Thus, the strategy that utilizes the link more productively will give higher total throughput.

Fig. 3.6: Normalized Throughput for Single Link Scenario, ($\alpha=\beta=\gamma=1$).

Fig. 3.6 illustrates the throughput achieved with the different strategies. ‘No Preemption’ has the highest throughput because all the connections admitted will be

completed successfully without subjecting them to possible interruption by preemption. However, mission critical connections may be rejected in ‘No Preemption’ when the link is fully utilized. Among the preemption strategies, OP-Prio shows the best performance due to its capability to select the optimized combination of connections for preemption.

Fig. 3.7: Probability of Success for Single Link Scenario, ($\alpha=\beta=\gamma=1$).

Fig. 3.7 illustrates the overall probability of success for the same single link simulations. It is defined as the ratio of *completed* connections to the *total arrival*. This performance metric is to gauge the number of connections completed with respect to the throughput. As higher throughput may be achieved either by connections with high bandwidth or long service time, probability of success gives another perspective on how the existing connections are affected by preemption. A higher probability of success is desired as it disrupts fewer numbers of end users.

Since ‘No Preemption’ does not trigger preemption, it shows the highest probability of success. With preemption strategies, more than one connection may be preempted to accommodate a new request which will lead to a lower probability of success.

Fig. 3.8 and Fig. 3.9 show the probability of success for connections of priority 0 and priority 7, respectively. The probability of success for connections of priority p is defined as the ratio of completed connections to the total arrival, both of priority p . The results for ‘No Preemption’ in Fig. 3.8 and Fig. 3.9 are similar to the one at Fig. 3.7 because the link treats the connections indifferently, regardless of the priority levels. A higher priority connection may be rejected if the link is congested. However with preemption schemes, priority 0 connections are given the opportunity to tear down lower priority connections. In Fig. 3.8, the probability of success for priority 0 connections maintains above 0.95 when preemption scheme is used. Little difference is observed between the different preemption strategies. The probability value drops at higher arrival rates because as a larger number of priority 0 connections are admitted and leading to congestion, future priority 0 connections may be rejected due to the fact that a priority 0 connection cannot preempt another priority 0 connection.

In Fig. 3.9, it is observed that ‘No Preemption’ maintains high probability of success whereas the rest of the schemes degrade the performance significantly. In order to minimize service disruption, a good preemption scheme should produce higher probability of success for lower priority connections. The results indicate that all the schemes are able to provide resources to the higher priority connections but ‘OP-Prio’ performs the best in minimizing service disruption, followed by ‘MHA’ and ‘Heuristics’. However, ‘OP-Prio’ uses more computational power due to its binary optimization solution.

Fig. 3.8: Probability of Success for Priority 0 Connections, ($\alpha=\beta=\gamma=1$). Fig. 3.9: Probability of Success for Priority 7 Connections, ($\alpha=\beta=\gamma=1$).

It is observed that ‘MHA’ performs better than ‘Heuristics’ for both the throughput and probability of success. However, loss of throughput still occurs because the preempted connections are not re-routed. Subsequent sections explore methods to minimize this loss of throughput as well as reducing service disruptions by implementing soft preemption. MHA will be used as the main framework for the subsequent formulations of both centralized and decentralized preemption strategies.

3.4 Centralized Preemption Strategy

The duration of service disruption caused by preemption can range from several minutes to hours if the connections cannot be re-routed. The end users’ perception on the network service provided will be affected. Furthermore, the network with preemption system will observe lower probability of success, which is the ratio of completed connections to the total arrival. This is due to the fact that more than one existing connections are likely to be preempted at each preemption instance. Thus, it is worthwhile to minimize service disruption as end users will have greater satisfaction and the network throughput will be improved. Fortunately, re-routing before tearing down is made possible by soft preemption [88]. The steps involved in the soft preemption [88] scheme are outlined as follows:

Soft Preemption Algorithm:

- Step 1:* Select the combination of connections to be preempted by using any preemption algorithm.
- Step 2:* Search for alternative feasible paths for the connections to be preempted.

Step 3: Set up alternative paths if found and switch the existing connections to the alternative paths.

Step 4: Release the resources on the preemption link and tear down the connections that cannot be re-routed.

Step 5: Set up the new request.

Existing preemption strategies [80-87] do not have the capability to minimize service disruption. Although soft preemption can be implemented as an add-on feature to these schemes, the information of which connection can be re-routed is not used. In the newly proposed centralized preemption strategy, this weakness is remedied. The MHA preemption algorithm is used as the fundamental framework. A brief outline of the centralized preemption algorithm is given as follows,

Centralized Preemption Algorithm:

Step 1: If a new connection triggers preemption, construct a subgraph $G' = (N', E')$ that excludes preemption link.

Step 2: Check whether the existing connections are re-routable on the subgraph G' . Mark the re-routable connections as they are more favorable as preemption choices.

Step 3: Find a path for the new connection using the minimum cost path, where the cost is proportionate to the connections (bandwidth) preempted.

Step 4: MHA selects the re-routable connections identified in *Step 2* for preemption.

Step 5: If the sum of bandwidth from the re-routable connections is less than the bandwidth required, use MHA to preempt some non re-routable

connections.

Step 6: Soft-preempt the selected connections if possible, or terminate by force.

The main objective of the centralized preemption algorithm is to take advantage of the re-routability of existing connections to minimize service disruption. *Step 1* of the centralized algorithm constructs a subgraph that excludes the preemption links such that existing connections are not re-routed on the same links. This will avoid the conflict of resources with the new connection. Let's denote the set of possible arcs that can be used to route the new connection as $\mathbf{L} \subseteq E$, thus the subgraph $G' = (N', E')$, $G' \subseteq G$, $N' \subseteq N$, and $E' \subseteq E$ is given by,

$$E' = E - \{e \in E: e \text{ the arcs used in } \mathbf{L} \text{ that trigger preemption}\}. \quad (3-16)$$

Fig. 3.10: Construction of Subgraph.

The network can use constraint-based routing such as delay-bound, hop-count constraint and k -shortest paths to control the size of $\mathbf{L}$. The illustration of G' construction is given in Fig. 3.10.

Given G' , the network can investigate to ascertain if the existing connections are re-routable as in *Step 2*. This is a greedy process where every connection on the preemption links with priority levels lower than the new connection will be checked. Let us denote a binary variable w as the *re-routing index*, which indicates whether a connection from the set $\mathbf{M}_K^{ij}$ can be re-routed. For all $c_k^{ij} \in \mathbf{M}_K^{ij}$ on link E_{ij} ,

$$\mathbf{W}_K^{ij} = \{w_1^{ij}, w_2^{ij}, \dots, w_K^{ij}\}, \quad (3-17)$$

$$w_k^{ij} = \begin{cases} 1 & \text{if } c_k^{ij} \in \mathbf{M}_K^{ij} \text{ can be re - routed ,} \\ 0 & \text{otherwise .} \end{cases} \quad (3-18)$$

This information is fed into the preemption algorithm and the connections in $\mathbf{W}_K^{ij}$ with $w_k^{ij} = 1$ will become the primary preemption choice.

To minimize service disruption, the author proposes that preemption links with the highest number of re-routable connections are selected ahead of other links that have to terminate existing connections. Let S_w be the sum of the bandwidth from existing re-routable connections on E_{ij} ,

$$S_w = \sum_{k=1}^K b_k^{ij} I_\psi(b_k^{ij}), \quad (3-19)$$

$$I_\psi(b_k^{ij}) = \begin{cases} 1 & \text{if } b_k^{ij} \in \psi, \\ 0 & \text{if } b_k^{ij} \notin \psi. \end{cases} \quad (3-20)$$

$$\psi = \{c_k^{ij} : w_k^{ij} = 1; c_k^{ij} \in \mathbf{M}_K^{ij}\}, \quad (3-21)$$

where $I_{\psi}(b_k^{ij})$ is an indicator function for the event ψ that c_k^{ij} is re-routable. The cost of the network link E_{ij} is denoted as θ_{ij} where,

$$\theta_{ij} = \begin{cases} 1 & \text{if } b_{new} \leq S_w + U_{ij} , \\ b_{new} - S_w - U_{ij} & \text{if } S_w + U_{ij} < b_{new} \leq U_{ij} + T_{ij} , \\ \infty & \text{otherwise .} \end{cases} \quad (3-22)$$

The cost of a network link will be equal to 1 if the bandwidth requested by a new connection can be fully satisfied by the the sum of idle bandwidth and re-routable bandwidth, $U_{ij} + S_w$. If the new connection needs higher bandwidth than that provided by idle and re-routable bandwidth, the network link cost is proportionate to the amount of excess bandwidth required. Thus, when the network is not able to find a path with all the links $E_{ij} \in l$ satisfying $U_{ij} \geq b_{new}$, the link cost (3-22) will be used to search for the appropriate path with preemption. One would like to find a path l such that the total link cost is minimized. The problem to find the minimum cost path is defined as follows,

Routing to Minimize Preemption (RMP)

$$\text{Min } \sum_{E_{ij} \in E'} \theta_{ij} x_{ij} \quad (3-23)$$

Subject to:

$$\sum_{j \in N'} x_{ij} - \sum_{j \in N'} x_{ji} = \begin{cases} 1 & \text{for } i = s , \\ 0 & \text{for all } i \in N' - \{s, t\} , \\ -1 & \text{for } i = t . \end{cases} \quad (3-24)$$

$$x_{ij} \in \{0, 1\}. \quad (3-25)$$

The binary variable x_{ij} represents the network flow from the source s to destination t . The first constraint (3-24) in RMP defines the flow conservation. The path chosen essentially favors the links that do not trigger preemption or all the preemptable connections are re-routable. If this condition cannot be met, it will select the links that terminate a smaller number of existing connections.

Once a path is identified, MHA will be used to select the connections based on the link condition as in *Step 4* and *Step 5*. If $U_{ij} < b_{new} \leq S_w + U_{ij}$, not all of the re-routable connections will be preempted. One can use MHA to determine the set of connections from $\mathbf{W}_K^{ij}$ with $w_k^{ij} = 1$ such that $\mathbf{z} \cdot \mathbf{b}^T \geq V$. Otherwise if $b_{new} > S_w + U_{ij}$, one can first preempt all the re-routable connections before using MHA to select the right combination from the rest of the preemptable connections.

After the path is chosen, the preemption link will start the re-routing process by sending re-routing requests to the source nodes. However, the alternative paths chosen by these to-be-preempted connections may interfere with each other which may lead to some of the connections being unable to utilize the alternative path. In such a case, the connections that failed to be re-routed will be terminated. No preemption will be allowed during this re-routing process so that network stability is not affected. In this centralized preemption strategy, a centralized depository and central controller are required to obtain the re-routing index. The centralized depository will keep track of the existing connections and the network links used to provide the resources. The information will be used by the central controller to obtain the re-routing index. When the information is available, it is sent to the individual links and preemption decision is made independently at each link. Among the possible candidates for the central controller are bandwidth brokers suggested in [91].

3.4.1 Computational Complexity

The heuristic algorithm proposed in [81] is upper bounded by $O(k^2)$, given that there are k number of connections on link E_{ij} that can be preempted. In order to decide the re-routing index, one needs to find routes with bandwidth and hop-count constraints and the routing complexity is given by $O(h * E)$ [1] where h is the hop count and E is the number of arcs in the network. Therefore, the computational complexity is given by $O(k^2 + k * h * E)$. Although extra computation is needed in this centralized preemption strategy, the computational complexity is not adversely affected and thus still upper bounded by $O(k^2)$.

3.5 Decentralized Preemption Strategy

The main challenge facing networks in the implementation of the centralized preemption strategy is the determination of the re-routing index. Networks need to update the network graph and attempt to re-route the existing connections to find out the re-routing index before deciding the connections for preemption. The implementation complexity of this problem motivated the author to develop a decentralized preemption strategy that minimizes the required information. In the decentralized preemption strategy, one does not have a central controller to collate the re-routing index from existing connections. The network link has no information as to whether the existing connections can be re-routed. Therefore, instead of selecting the links that have the higher number of re-routable connections, we select the path that minimizes the amount of bandwidth that needs to be preempted. A brief description of the decentralized algorithm is given below,

Decentralized Preemption Algorithm:

- Step 1:* Route the new connection on the path that minimizes bandwidth to be preempted.
- Step 2:* On the preemption links, select connections for preemption from the lowest priority if $S_p \leq V$.
- Step 3:* At the priority level p when $S_p > V$, check whether connections at priority p are *locally* re-routable.
- Step 4:* Use MHA to select the locally re-routable connections at p .
- Step 5:* If the sum of bandwidth from the locally re-routable connections is less than the bandwidth required, use MHA to preempt some non re-routable connections.
- Step 6:* Soft-preempt the selected connections, starting from the higher priority and locally re-routable connections.

In order to minimize the bandwidth preempted, one can assign the link cost θ_{ij} as follows,

$$\theta_{ij} = \begin{cases} 1 & \text{if } b_{new} \leq U_{ij} , \\ b_{new} - U_{ij} & \text{if } U_{ij} < b_{new} \leq U_{ij} + T_{ij} , \\ \infty & \text{otherwise .} \end{cases} \quad (3-26)$$

When all the possible paths with sufficient bandwidth cannot be found, equation (3-26) can be used to find the minimum cost path (i.e. similar to RMP). Hence, the number of existing connections that have to be preempted is reduced and the existing connections will have higher chances of being re-routed, due to less competition for network resources.

In [88], when soft preemption is triggered, the congested link will send a re-routing request signal to the source node of the existing connection. Upon receiving

it, the source node will start searching for an alternative path that not only has sufficient idle bandwidth but must also be link disjoint with the congested links. Such an alternative path may be found if the removal of the congested links does not result in a disconnected graph between the source and the destination. Thus, if the existing connection can be locally re-routed between the two end nodes of the congested link in the graph G' , it is possible that it can be source re-routed to its destination. Before triggering preemption, the congested link will investigate whether the existing preemptable connections can be locally re-routed. Similar to the centralized algorithm, a connection will be assigned a local re-routing index, $w_k^{ij} = 1$ if it is locally re-routable and $w_k^{ij} = 0$ otherwise. Fig. 3.11 depicts two possible locally re-routable paths for a congested link.

Fig. 3.11: Local Re-routing by the Congested Link.

The preemption link will then preempt the existing connections from the lowest priority. At the priority level p such that $S_p > V$, the link will preempt the connections with $w_k^{ij} = 1$ before using MHA to select the rest, if necessary. This algorithm is designed to safeguard the integrity of the priority level because in a decentralized strategy, the network links do not possess the complete re-routing information. A connection that is locally re-routable on one congested link may not be locally re-routable on the others. Furthermore, the local re-routing paths chosen by the existing

connections may interfere with each other. Therefore, it is more sensible to satisfy the priority level before choosing the locally re-routable connections.

After making the preemption decision, the network link will now have a list of connections to be preempted. Soft preemption is implemented by choosing the connections with $w_k^{ij} = 1$ from the highest priority. This is to ensure that given the preemptable connections, the higher priority ones are provided with a better service by being re-routed first. Hence, this strategy will help the network link to achieve both the objectives of satisfying priority level and minimizing service disruption. Since preemption decisions are made on links independently, preemption is triggered successively from one congested link to the other on the path (e.g. from upstream to downstream). This allows the downstream-congested links to free some of the occupied bandwidth if the upstream links and downstream links are servicing the same preemptable connections. If the freed bandwidth is higher than the bandwidth requested, no preemption is needed.

The primary advantage of the centralized strategy lay in its capability to acquire network-wide re-routing information and to utilize that information to minimize service disruption. On the contrary, the decentralized strategy only has partial re-routing information available to it in the form of the outcome of the local re-routing strategy. If only one link of the path chosen will trigger preemption, the existing connections with $w_k^{ij} = 1$ will be re-routed successfully. However, as the number of congested links increase, connections that can be locally re-routed on one congested link may not be re-routable on the other. This will inevitably affect the performance of the decentralized strategy. However, with less information to process and collate, the decentralized strategy will respond faster to users' requests. Furthermore, it will

be more easily deployable as the links only need to manage their local information. Performance comparisons between the centralized and decentralized strategies are presented in the next section.

3.6 Simulation Results and Discussions

Fig. 3.12: Network Topologies.

Simulation experiments written in C++ are carried out to analyze the performance of the proposed preemption scheme as compared to the existing schemes. In this work, we carry out extensive performance analysis by executing simulations on 4 different topologies, i.e. a small grid network, NSF-net, three medium size random grid networks with 18 nodes and three random grid networks with 30 nodes. The random grid networks are generated based on the model proposed in [92]. The network topologies are given in Fig. 3.12.

For each network topology, each link is assumed to have a bandwidth equals to 500 capacity units. Every single node in the network can be a source node or destination node. Poisson arrival process is assumed and the connection holding time is exponentially distributed with mean 800 seconds. The bandwidth request is uniformly distributed between $U(10 - 50)$ capacity units with priority level uniformly distributed from 0 to 3. The coefficients for MHA, i.e. β and γ , are set to 1 (Recall that α is not needed as MHA algorithm preempts from the lowest priority). Every new connection will trigger preemption when congestion happens. Comparisons are made between the non-preemption system, centralized preemption, decentralized preemption and MHA without re-routing. The simulation results are obtained for each of the topologies. The following results presented are extracted from a random grid network with 30 nodes. The same trends of results are observed for different topologies.

Fig. 3.13 illustrates the normalized throughput achieved by the different strategies. Throughput is defined as the dot product of bandwidth assigned and service time and it is accrued only when a connection completes successfully. This performance metric is important because the main design criterion is to minimize service disruption, which in turn would improve network throughput as a smaller number of connections are terminated. Note that the network without any preemption scheme has the highest throughput because existing connections are never prematurely terminated. For networks with a preemption system, a preemption event on a congested link may disrupt multiple connections causing lower throughputs. The centralized strategy performs better than the decentralized strategy as the network has complete information regarding the re-routing of existing connections. By choosing

Fig. 3.13: Network Normalized Throughput.

re-routable connections for preemption, the network can reduce the degradation of throughput. The preemption system without re-routing capability will seriously affect the network performance. In Fig. 3.13, network throughput peaks at the arrival rate of (0.2 connection/s), beyond which the performance of decentralized strategy and that without re-routing degrade gradually. This indicates that as the network is subjected to high arrival rates, intensive preemption events will occur and affect the completion of existing connections. The preemption system therefore appears to be more appropriate for arrival rates of medium to high degree.

The probability of success in Fig. 3.14 is defined as the ratio of connections completed successfully to the total connection arrivals. Notice that the centralized strategy performs marginally better than no preemption system at low arrival rate because the network is able to accommodate more connections as most of the existing

Fig. 3.14: Probability of Success.

connections preempted can be re-routed. However, the throughput achieved by the centralized strategy (Fig. 3.13) is lower because if a connection is not re-routed, its throughput will not be accrued. The decentralized strategy performs better than a preemption strategy with no rerouting (i.e. ‘No Reroute’) as some of the connections get re-routed and are completed successfully in this case.

In Fig. 3.15, the *service disruption probability* is defined as the ratio of connections terminated by force without re-routing to the sum of connections uninterrupted throughout their lifetime. This performance metric gives a measurement on how likely a lower priority connection is terminated prematurely. The centralized preemption strategy has the lowest service disruption rate because of its capability to select the connections that can be re-routed. Although the decentralized strategy is equipped with incomplete information, the re-routing index

Fig. 3.15: Service Disruption Probability.

is still able to help the links select the re-routable connections. As may be expected, the approach with no re-routing shows the worst performance here as none of the preempted connections are re-routed. With lower service disruption, one would expect end users will have greater satisfaction with the services provided.

The probability of re-routing in Fig. 3.16, which shows the ratio of the number of connections re-routed to the sum of connections selected for preemption, decreases gradually as the arrival rate increases. Fig. 3.15 confirms our observations where the centralized strategy achieves lower service disruption rate with higher re-routing probability. On the average, the decentralized strategy shows approximately 5% poorer performance than the centralized strategy.

Fig. 3.16: Probability of Re-routing.

Fig. 3.17: Percentage of Call Completion for Decentralized Strategy.

The percentage of call completion for the individual priority levels in the decentralized strategy is illustrated in Fig. 3.17. Recall that the priority levels are uniformly distributed between 0 and 3. Therefore, one-fourth of the total connections are from priority 0, and the same applies to priority 1, 2 and 3. The percentage of call completion is used to capture the level of service degradation suffered by lower priority connections when preemption is implemented. For a network without preemption scheme, the percentage of call completion for all the individual priority levels are clustered tightly around the horizontal line of 25% throughout all the different arrival rates. This is due to the fact that no preference (i.e. by preempting lower priority connections) is given to the higher priority connections when the network is congested.

In Fig. 3.17, when at low arrival rates, each of the four priority levels shows 25% completion as preemption events are less frequent. However, with increasing arrival rates, the network is biased towards high priority connections as lower priority connections tend to get preempted more often. It is observed that connections of priority 3 (lowest level) suffer the worst degradation as these are the ones which are primarily preempted. Similar trends are also observed for the centralized preemption strategy. For no re-routing scheme as shown in Fig. 3.18, lower priority connections (i.e. priority 2 and 3) experience greater service degradation as compared to the decentralized strategy with re-routing because many of the disrupted connections are not re-routed to preserve the ongoing services. As a result, higher priority connections make up the majority of the connections completed.

Fig. 3.18: Percentage of Call Completion for Decentralized Strategy with No Re-routing.

3.7 Simulations on Non-Uniform Links

This section studies the performance of the proposed algorithms on a network with non-uniform links. The network topology of 30 nodes introduced earlier in Fig 3.12(c) is shown in Fig. 3.19 with the links drawn with thicker lines are bottleneck links. The bottleneck links are defined as the links that observe more than 70% of utilization which cumulatively span for more than 70% of the simulation time. The bottleneck links are assigned with 750 capacity units with the rest having 500 capacity units. All other parameters remain the same values as in the previous section.

Fig. 3.19 : Network Topology with Non-uniform Links.

Fig. 3.20 : Normalized Throughput for Non-Uniform Links.

The normalized network throughput is illustrated in Fig. 3.20. In general, it is observed that it shows the same trend as in Fig. 3.13 where all uniform links are used. However, the throughput achieved here is higher because of the added capacity at the bottleneck links. We also notice that throughput stabilizes at the arrival rate of 0.28

connection/s. Beyond this arrival rate, congestion builds up at the rest of the links limiting the increase of the network throughput.

On the other hand, service disruption probability (Fig. 3.21) is reduced significantly as well due to the extra capacity. The service disruption probability stabilizes gradually at arrival rates higher than 0.36 connection/s as the new connections then find it harder to trigger preemption since the scenario is such that existing connections are dominated by high priority ones.

Fig. 3.21 : Service Disruption Probability for Non-Uniform Links.

3.8 Summary

Although preemption can provide resources for new incoming high priority connections, the network suffers the drawback of higher service disruption probability and lower network throughput. In order to minimize service disruption,

centralized and decentralized strategies are proposed to preempt the connections that can be re-routed in order to implement soft preemption more effectively. In the centralized strategy, the service disruption is greatly reduced with a concurrent improvement in the network throughput. Although the decentralized strategy is not as effective as the centralized strategy, it is much more easily deployable since only local information is needed. For networks with light loading, the author envisages that the decentralized strategy will be able to provide the benefits of accommodating high priority connections without adversely affecting the overall network performance. With higher loading, its performance is still significantly enhanced as compared to the preemption system without re-routing.

Although both the centralized and decentralized schemes are able to improve network throughput as compared to existing schemes, the improvement is achieved with the higher cost of re-routing. In the re-routing process, the network needs to search for the alternative paths, set up the selected path, switch the existing connection to the new path and finally tear down the existing path. Excessive re-routing may affect network stability and introduce extra delay in the setting up of the new higher priority connection. Hence, preemption scheme is more appropriate for networks with medium loading.

4. Path Selection with Preemption and Re-routing Control for MPLS

Multi Protocol Label Switching (MPLS) networks enhance the services of conventional best-effort IP networks by providing an end-to-end Quality of Service (QoS) guaranteed Label Switched Paths (LSP) between customer sites. The LSP has to be set up in advance before carrying the traffic. Contention for network resources may happen if many LSPs try to use a common network link with limited bandwidth. Preemption scheme may be used such that the limited network resource is given to the higher priority LSPs. It is observed that preemption is a practical problem which can arise from events such as oversubscription, node failure and link failure. In Chapter 3, preemption algorithms with re-routing were developed in order to provide resources to higher priority connections without seriously affecting the services of lower priority connections. Without loss of generality, this chapter investigates the problem of path selection and preemption in oversubscribed networks. The strategies formulated can be readily applied to the cases of node failure and link failure by simply considering the affected LSPs as new requests. The objective of the proposed strategies in this chapter is similar to Chapter 3, i.e. to provide better resources for higher priority LSPs and minimize service disruption to lower priority LSPs.

However, this chapter differs by incorporating more active control mechanisms such as path selection and re-routing control into the preemption schemes.

Given an MPLS network with existing traffic matrices, a newly arrived high priority LSP may find L possible paths between its source and destination. It may select the shortest path which may trigger preemption or choose a longer path which however utilizes more resources. The preemption strategies are initially formulated with global re-routing (as shown in Fig.2.9), where the source node selects the best path to the destination. This investigation includes the effects of routing of higher priority LSPs on the shortest path and its alternative paths. It is shown in later sections that by persistently routing the higher priority LSP on the shortest path, more preempted LSPs can be re-routed which would increase the network throughput and minimize service disruption. Since excessive re-routing may degrade the network performance, a re-routing control strategy is proposed to constrain the length of these alternative paths. Finally, a decentralized preemption strategy with local re-routing is also presented to approximate the performance of the proposed strategy with significantly lower control overheads. Simulation results show that network throughput is greatly improved as compared to existing preemption strategies [80–87].

4.1 Network Model

The network model used in this formulation is similar to the one defined in Chapter 3, Section 3.1. However, for clarity purposes, some of the definitions are restated in this section. The network is represented by the graph, $G = (N, E)$ where N is the set of all nodes and E is the set of all edges. The total number possible node pairs is given by

$N(N-1)/2$. The set of all node pairs is represented by D , each indexed by d with source s and destination t . For an edge $E_{ij} \in E$, its bandwidth capacity is given by CAP_{ij} . MPLS can support up to eight priority levels ranging from numerical value 0 to 7, with 0 represents the highest priority. Let us denote the total number of priority levels supported as P with each priority level as p , where $p = (0, 1, 2, \dots, P-1)$.

Fig. 4.1: Source-destination Node Pair and Possible Paths.

A new LSP, c_{new} that requests network resources from the MPLS network will notify its bandwidth requirement b_{new} , priority level p_{new} , and source-destination node pair d_{new} . The following notation represents the new LSP, $c_{new} = (b_{new}, p_{new}, d_{new})$. For a given node pair d , there are L numbers of possible paths through the network, each indexed by l , as illustrated in Fig. 4.1. With the functionality of MPLS-TE [17], the network can explicitly choose the favorable path and the path chosen is not necessarily the shortest path. We order L in accordance to the ascending order of number of hops. Therefore, the first path in L is always the shortest path. The pair (d, l) identifies the l th path for the node pair d . The residual bandwidth on edge E_{ij} is represented by U_{ij} , $U_{ij} = CAP_{ij} - S_{ij}$, where S_{ij} is the sum of bandwidth used by the existing LSPs on E_{ij} . The network is able to admit LSP c_{new} on route (d_{new}, l) if the following condition is satisfied.

$$U_{ij} \geq b_{new}, \text{ for all } E_{ij} \in (d_{new}, l) \quad (4-1)$$

If the condition (4-1) is violated, LSP c_{new} can choose to exhaust all the options available in $\mathbf{L}$ or trigger preemption to select the favorable path. This reveals a tradeoff between the selection of favorable path but at the expense of preemption and the selection of the less favorable path. Let T_{ij} denotes the preemptable bandwidth on edge E_{ij} , which is the sum of bandwidth from existing LSPs with priority level lower than the new LSP as given in (3-4). The path (d_{new}, l) can be used to route LSP c_{new} by implementing preemption only if all the edges satisfy the following condition,

$$U_{ij} + T_{ij} \geq b_{new}, \text{ for all } E_{ij} \in (d_{new}, l) \quad (4-2)$$

Preemption algorithm is used to determine the combination of LSPs to be preempted at the congested links in order to route the new request. For the purpose of minimizing the service disruption, soft preemption is introduced in [88] to re-route the LSPs to be preempted. Soft preemption functions such that alternative paths are established and the ongoing traffic is switched to the alternative paths before the existing LSPs are preempted. Although end users may notice a short period of delay caused by the execution of soft preemption, it does not affect the throughput and services provided. However, not all preempted LSPs can be re-routed through soft preemption, a loss of network throughput will occur on LSPs that cannot execute soft preemption successfully.

Given that there are $\mathbf{L}$ possible paths that can be used to route the new LSP c_{new} , the network can choose the path that will not trigger preemption or path that triggers preemption on one or more edges. Since the $\mathbf{L}$ possible paths are ordered with respect to the increasing number of hops, the new LSP will find that it is using more network

resources as it proceeds through the path search. In order to conserve network resources, the new LSP will choose the shortest path if it has sufficient bandwidth. However, if the shortest path is not available, the LSP can trigger preemption to acquire the shortest path or choose a longer path but risk blocking more future LSPs. This consideration is particularly important to higher priority LSPs because only these LSPs can obtain enough preemptable bandwidth from congested link, and if these are routed on longer paths, they cannot be preempted by other LSPs. Next section investigates the effects of these issues on the network performance.

The majority of existing strategies [80-87] seek to exhaust all the possible paths before choosing the one that minimizes the number of links that trigger preemption. However, simulation results in the later sections show that network throughput is improved if higher priority LSPs are constantly routed on the shortest path even though preemptions are triggered occasionally.

4.2 Preemption and Re-routing Strategy

Two variations of strategies are investigated in this section. The first strategy lets the new LSP searches for all L possible paths before preemption is triggered. The second strategy explores the effects of constantly selecting the shortest path for the higher priority LSP even it is at the expense of preemption. Given L possible paths of the node pair d , and denote the hop-count difference between route $(d, l+1)$ and (d, l) as $\Delta h_{l+1, l}$.

$$\Delta h_{l+1, l} = H(d, l+1) - H(d, l) \quad (4-3)$$

The function H is used to compute hop counts of the pair (d, l) . As the L possible paths are ordered in the ascending order of hop count, $\Delta h_{l+u,l} \geq 0$ for all integer $u > 0$. Since preemption is only triggered when (4-1) is violated, it provides us the clue that, at this point, the network load is possibly within the medium to high range. The study in [93] shows that a routing algorithm that limits the hop count (i.e. shortest path) performs better than a load balancing routing algorithm in networks which are highly loaded or overloaded. This insight proves useful in the design of the preemption strategy in this chapter.

4.2.1 Search All and Preempt (SEP)

In this strategy, the new LSP with node pair d (regardless of its priority level) searches all the L possible routes for d in order to find the path that satisfy (4-1), i.e. all the edges on the path have sufficient bandwidth to admit the new LSP. The search is stopped immediately if a path is identified. Further search is not necessary as the subsequent paths will be of the same or higher length. However, if none of the L possible paths can admit the new LSP, preemption will be triggered at one of the routes to acquire the needed resources. A search will be carried out to find the route with the least number of edges that trigger preemption so that service interruption is minimized. If two or more paths are identified, the path is chosen arbitrarily. The rationale for SEP is to avoid preemption so that the services of ongoing LSPs are not interrupted. The new LSP will be rejected if none of the route can satisfy (4-2). SEP represents the common path selection approach in the existing preemption schemes [80-87].

A simple illustration of SEP is given in Fig. 4.2. Given a new LSP to be routed between the node pair $d = (s, t)$, the network finds that the shortest and the second shortest paths are congested. SEP ensures that all the possible paths are searched and in the case of Fig. 4.2, the new LSP is routed on l_3 which has sufficient resources.

Fig. 4.2: An Illustration of SEP Algorithm.

4.2.2 Limit Hop Count and Preempt (LIP)

This strategy limits the search of possible paths and allows the use of preemption to attain the shortest path. This is more applicable to higher priority LSPs because low priority LSPs will find it harder to acquire preemptable bandwidth and thus have lower successful probability of preemption. By constantly selecting the shortest path for the higher priority LSPs, these LSPs are less likely to interfere with future requests. A higher priority LSP (e.g. priority 0) that is routed on a longer path not only will block future requests, it cannot be preempted even by the other high priority LSPs. Since network throughput will be affected, this strategy limits the path search by a threshold value, $\Delta h_{l,1} \leq \tau$, e.g. by assigning $\tau = 1$, the initial search space only covers paths that have at most one hop more than the shortest path. The search is stopped if a path with sufficient residual bandwidth is successfully found; otherwise the path with the least number of edges that trigger preemption will be selected. If none of the paths is feasible, the search space is expanded by one hop at a time until

the search space covers all the L possible paths. This search space extension is necessary to preserve the integrity of priority levels. By adjusting the threshold τ in accordance to different priority levels, the strategy allows higher priority LSPs to route on the shortest path and lower priority LSPs on relatively longer paths.

In Fig. 4.3, the shortest path is $l_1 = (s, t)$. The difference in hop count between the shortest path and the rest of the paths is given as such, $\Delta h_{2,1} = 1$ and $\Delta h_{3,1} = 2$. If the network assigns $\tau = 1$, only paths l_1 and l_2 will be considered in the initial search. However, if both l_1 and l_2 are congested and do not have sufficient preemptable bandwidth, the search space will be expanded by one hop count at a time, i.e. l_3 will be considered next. Given that if $\tau = 1$, the network will try to trigger preemption to route the new LSP. Since l_1 contains only one congested link and assuming that it has sufficient preemptable bandwidth, it will be selected by the algorithm.

Fig. 4.3: An Illustration of LIP Algorithm.

4.2.3 Re-routing Control

On the edge $E_{ij} \in (d_{new}, l)$ that triggers preemption, we propose that soft preemption is executed in order to minimize service disruption. Unlike link fault or node failure that triggers MPLS protection, preemption is more closely related to MPLS management that oversees LSP competition. No immediate tearing down of LSPs is

necessary. Therefore, a grace period is allowed whereby existing LSPs are re-routed. In this re-routing process, the edge on which the preemption is triggered will send a re-routing signal to the source node of the to-be-preempted LSP one at a time. The source node is responsible to find an alternative path which does not interfere with the path used by the new LSP so that no competition of resources happens on all $E_{ij} \in (d_{new}, l)$. Global re-routing approach is preferable as the source node could find the shortest possible alternative path to destination.

As the network has no information whether a LSP can be re-routed successfully, the re-routing process is started from the LSP with the highest priority among the to-be-preempted LSPs until the grace period expires. In order to limit the network resource consumed by the re-routed LSP, a new threshold value σ is introduced so that the hop count of the alternative path (d, l_q) does not exceed that of the original path (d, l) by more than σ hops.

$$H(d, l_q) - H(d, l) \leq \sigma, (d, l_q) \neq (d, l) \quad (4-4)$$

An illustration of the re-routing control is shown in Fig. 4.4. In the example, the new LSP, c_{new} , is routed on the shortest path l_1 . However, the network needs to trigger preemption because the idle bandwidth, U_{st} , is zero. The four existing LSPs on l_1 , i.e. c_1 to c_4 , are given in Fig. 4.4. In order to preserve the integrity of priority levels, the network will preempt from the lowest priority. In this case, LSPs c_2 , c_3 and c_4 are able to provide sufficient bandwidth (22 units) to the new LSP (20 units). Subsequently, re-routing process will begin with the LSP that has the highest priority among the preemptable LSPs. Hence, c_2 will be re-routed first and followed by c_3 and c_4 .

Assuming that $\sigma = 1$, i.e. the alternative paths chosen for re-routing cannot be longer than the shortest path by one hop count, c_2 is re-routed on l_2 and no LSPs can be re-routed on l_3 . However, if $\sigma = 2$, the re-routing will proceed such that c_2 is re-routed on l_2 and c_3 is re-routed on l_3 . LSP c_4 will be terminated due to insufficient bandwidth on the alternative paths.

Fig. 4.4: An Illustration of Re-routing Control.

4.2.4 Congestion Based Preemption Algorithm

After the re-routing process, the amount of residual bandwidth will be higher as some of the existing LSPs are re-routed. If $U_{ij} \geq b_{new}$, no tearing down of LSP is required, otherwise we will have to tear down the appropriate combination of LSPs to free up sufficient bandwidth. Since preemption mainly occurs when the network load is high, the author proposes that the LSP on the most congested link or the one that consumes more network resources should be terminated. This will help to ease the congestion level of the network and increase the probability of accepting future LSP without triggering preemption. The congestion level of LSP c_k with (d, l) is defined as

$$\lambda_{c_k, (d, l)} = 1 - \frac{\arg \min_{E_{ij} \in (d, l)} U_{ij}}{CAP_{ij}} \quad (4-5)$$

The score function that is used to evaluate the LSPs for preemption is given by (4-6). It considers the hop-count difference between the currently used l th path and the shortest path, and also the congestion level of the LSP. For existing connection c_k ,

$$F_s(c_k) = \omega_1 \cdot \Delta h_{l,1} + \omega_2 \cdot \lambda_{c_k(d,l)} \quad (4-6)$$

ω_1 and ω_2 are the associated weights. However, in order to satisfy QoS requirements, only LSPs with priority lower than the new LSP are preemptable. On the edge $E_{ij} \in (d_{new}, l)$ that triggers preemption, let S_p denotes the total bandwidth of the existing LSPs at priority level p . All the LSPs at the lowest priority, $P-1$, will be preempted if $S_{P-1} < b_{new} - U_{ij}$. This process will continue to the LSPs with the next priority level until $S_q > b_{new} - U_{ij}$, where only a number of existing LSPs at priority q will be preempted. Thereafter, the score function (4-6) is used to determine the combination of LSPs to be terminated. The algorithm will preempt with the descending order of the score function until bandwidth requirement is satisfied. This essentially means that the LSP that uses relatively more resources than the shortest path and occupies the most congested link will be preempted. Hence, the network resource consumption and the congestion level are reduced.

Section 4.2.2 – 4.2.4 introduce a number of parameters, i.e. σ , τ , ω_1 , and ω_2 , for the distinctive tasks of path selection, re-routing control and preemption, respectively. In the subsequent sections, these parameters are investigated individually using simulations written in C++.

4.3 Performance Evaluation

Fig. 4.5: NFS-NET Network Topology.

This section presents the performance of the proposed preemption strategy i.e. SEP and LIP against existing approaches. The effects of the threshold value, σ and τ are investigated. Simulations are carried out on the same network topology as used in [55] and shown in Fig. 4.5. The network topology consists of 15 nodes and 28 links with all the nodes can act as both source and destination. All the links are bidirectional with bandwidth capacity of 500 units. The network supports four priority levels from 0 to 3.

New LSP arrives at the network with randomly chosen source-destination pair. The bandwidth request and priority level are uniformly distributed with $U(10, 50)$ and $U(0, 3)$ respectively. LSPs arrive according to the Poisson arrival process and the service time is exponentially distributed with mean of 800s. The network performance achieved by varying the traffic arrival rate from 0.02 LSP/second to 0.3 LSP/second is investigated. In the simulation studies, the performance of the congestion based preemption algorithm is investigated first, followed by the re-routing control strategy and finally the combination of the different schemes.

Fig. 4.6: Throughput for No Preemption, PREM and SEP without Re-routing ($\omega_1=1$, $\omega_2=10$).

Fig. 4.6 examines the performance of SEP without re-routing against the preemption algorithm proposed in [81], hereby named as PREM. The main purpose here is to obtain the best possible values for ω_1 and ω_2 used in the congestion based preemption algorithm (Section 4.2.4) such that the network throughput is maximized. No LSP re-routing is executed in Fig. 4.6 so that the network performance relies entirely on the different preemption algorithms. Furthermore, SEP is chosen here because it is similar to the common path selection process used in the existing preemption strategies, including [81]. Parameters σ and τ are dropped because LIP scheme and re-routing control are not implemented. Multiple simulations were run by varying the values of ω_1 between [1, 2, ..., 5] and ω_2 between [5, 10, ..., 30]. In the right hand side of (4-6), the value of the first term is an integer that represents the hop-count difference between the current path used against the shortest path whereas

the second term is a fraction between 0 and 1. Based on the topology in Fig. 4.5, simulation results show that the best performance is achieved when $\omega_1=1$ and $\omega_2=10$, in which both criteria exert relatively similar impact on the overall function. By fixing $\omega_1=1$, ω_2 can be adjusted upward or downward ($\omega_2>10$ or $\omega_2<10$) depending on a bigger or simpler topology. A simple approach may set ω_1 to 1 and ω_2 to the average number of hop counts in the shortest paths for different node pairs.

In Fig. 4.6, both SEP and PREM perform worse than the network without preemption because possibly more than one LSP are terminated in order to admit a new request. Therefore, constant execution of preemption will degrade the performance of the network. However, without preemption, all the LSPs are treated equally regardless of its priority and thus a high priority LSP may find itself being rejected due to network congestion. SEP achieves higher throughput than PREM because it is able to ease the congestion level of the network by terminating the LSP that utilizes more resources and occupies the most congested link. This reflects the fact that preemption is in proportion to the network load and that higher performance can be achieved by easing the load.

Fig. 4.7 illustrates the probability of success for different preemption schemes without executing re-routing. Probability of success is defined as the ratio of LSPs completed to the sum of total LSPs arrival. E.g. the probability of success for priority 0 LSPs is the ratio of priority 0 LSPs completed to the total arrival of priority 0 LSPs. Without preemption scheme, the probability of success for priority 0 LSPs is similar to the overall performance because the network does not give preference to the high priority LSPs. For SEP and PREM, higher probability of success is achieved by preempting existing lower priority LSPs.

Fig. 4.7: Probability of Success for Prio 0 LSPs without Re-routing ($\omega_1=1$, $\omega_2=10$). Fig. 4.8: The Effects of σ on Network Throughput and Re-routing Probability ($\omega_1=1$, $\omega_2=10$).

The effects of the re-routing threshold σ on the network throughput and re-routing probability at the arrival rate of 0.3 LSP/second (high load) are presented in Fig. 4.8. Since only SEP scheme is used, the parameter τ is ignored. Re-routing probability is defined as the ratio of the number of LSPs re-routed to the total number of LSPs selected for preemption. As the re-routing threshold σ increases, the length of the alternative path allowed increases proportionately. The preemption strategy in Chapter 3 represents the case where σ is unlimited. Note that network throughput peaks when $\sigma = 3$ and it decreases gradually with higher σ . However, re-routing probability increases with respect to σ . This shows that excessive re-routing of preempted LSPs will not improve the network performance; in fact it will degrade performance if control overhead associated with the re-routing process is taken into consideration. The throughput achieved at $\sigma = 3$ in Fig. 4.8 is higher than the throughput of SEP in Fig. 4.6 because re-routing strategy allows LSP services to continue uninterrupted. However, with re-routing, more over-head operations will be involved such as the search for alternative paths and LSP switching.

The rest of the simulations set the following parameters such that $\sigma = 3$, $\omega_1=1$ and $\omega_2=10$. The threshold τ for LIP, is assigned in the following fashion to investigate its impacts.

Case A: ($\tau = 0$, prio 0), ($\tau = 1$, prio 1), ($\tau = 2$, prio 2).

Case B: ($\tau = 1$, prio 0), ($\tau = 2$, prio 1), ($\tau = 3$, prio 2).

Case C: ($\tau = 2$, prio 0), ($\tau = 3$, prio 1), ($\tau = 4$, prio 2).

Fig. 4.9: Network Throughput of LIP (Case A, Case B, Case C), SEP and No Preemption.

In Fig. 4.9, the throughput performances of LIP (Case A, Case B, Case C) show significant improvement over SEP. This shows that strictly controlling the length of paths chosen by high priority LSPs will improve the overall network performance. When high priority LSPs constantly use shorter paths, network resource consumption and its interference on future requests are both minimized and, hence, more LSPs can be admitted overall. The improvement achieved on throughput is about 15% higher as compared to non-rerouting preemption strategy such as PREM [81]. This suggests that preemption strategy with re-routing capability, i.e. soft preemption can reduce the disruptive nature of preemption. LIP even performs marginally better than ‘No Preemption’ at lower arrival rates where most of the preempted LSPs can anyway be re-routed.

Fig. 4.10: Re-routing Probability of LIP (Case A, Case B, Case C) and SEP.

The observation that LIP performs better than ‘No Preemption’ in terms of throughput at low arrival rates is confirmed by the high re-routing probabilities observed for LIP in Fig. 4.10 for the arrival rates between 0.1 and 0.2 LSP/second. However, SEP cannot perform better than “No Preemption” in terms of throughput at low arrival rate because the high priority LSPs may be routed on the longer paths and thus interfere with future requests.

Fig. 4.11 illustrates the Probability of Success for the different LIP cases and for SEP. Probability of Success is defined as the ratio of the total number of LSPs completed successfully to the total arrival. Interestingly, Case A and Case B perform similarly in terms of throughput and success probability although Case A clearly has higher re-routing probability. The reason is that by tightly controlling the path length of high priority LSPs with smaller values of τ , more preemptions are triggered thus

Fig. 4.11: Probability of Success for LIP (Case A, Case B, Case C) and SEP.

leading to a higher re-routing probability. These extra preemption events do not contribute to the network performance. It merely indicates the underlying frequent reorganization of LSPs, in which high priority LSPs are constantly trying to acquire the shortest path. In view of this, if we take re-routing overheads into consideration, then LIP Case B would be a better strategy as fewer LSP re-routings are required.

Fig. 4.12 shows the average path length versus priority levels as obtained by the three cases of LIP, the SEP and the ‘No Preemption’ strategies for LSPs completed successfully. The results presented indicate that LIP can reduce the path length significantly for all the four priority levels. Without preemption, the average path length across different priority level is almost the same. For LIP, the average path length increases from priority 0 to priority 1 but decreases abruptly thereafter (the drop happens at priority 3 in Case A). This is due to the fact that at high arrival rates

Fig. 4.12: Average Path Length of LIP (Case A, Case B, Case C), SEP and No Preemption.

(0.3 LSP/second), more resources are acquired by high priority LSPs causing low priority LSPs to be routed on longer path which are then subject to intensive preemption. Hence, only those low priority LSPs which get routed on the shortest path tend to be completed successfully. This is confirmed by the average path length of priority 3, which has about the same value as for priority 0. The average path length for SEP is higher than 'No Preemption' for priority 0 and priority 1. The main operation for SEP is to search for all possible paths and select the shortest path that has sufficient resources. If no such path with sufficient resource is found, SEP will trigger preemption on the path that has the least number of preemption links. However, there is not guarantee that the path chosen is the shortest one, in fact SEP will prefer a longer path with one preemption link than a shorter path with two

preemption links. Hence, with SEP's tendency to search for all possible, it increases the average path length of priority 0 and 1 LSPs.

Fig. 4.12 shows that LIP will be better at satisfying delay sensitive applications. The choice for the threshold value τ is highly dependent on the applications supported and the network performance to be achieved. If the application is highly delay sensitive, assigning $\tau = 0$ will ensure that high priority LSPs are routed on the shortest path. However, with all the overheads taken into consideration, LIP Case B seems to be able to provide the benefits of higher throughput and resource consumption minimization. It is a balance between the strict control of Case A and good overall performance.

Fig. 4.13: Distribution of LSPs Completed Successfully for No Preemption.

Fig. 4.14: Distribution of LSPs Completed Successfully for LIP Case B.

Fig. 4.13 and Fig. 4.14 show the distribution of the successfully completed LSPs as per their different priority levels. Since there are four priority levels, a network that does not support preemption will treat the LSP equally and the distribution will be 25% for each of the priority levels as shown in Fig. 4.13. Accordingly, in Fig. 4.14, note that the distributions are close to 25% for all the priorities at low arrival rates, i.e. below 0.14 LSP/second. This is because, in this region, most of the LSPs preempted can be re-routed. One can confirm this from Fig. 4.10 which shows that that the re-routing probability for LIP Case B in this region is better than 90%.

However, with higher traffic loads, the competition for resources increases proportionately and only LSPs with higher priority are successful. The increases observed for higher loads in the distributions for priorities 0 and 1 in Fig. 4.14 coincides with the degradation of performance observed for priorities 2 and 3. Note

that the distribution of priority 2 rises slightly between 0.18 and 0.22 LSP/second before falling gradually and going below the 25% benchmark when the LSP arrival rate exceeds 0.28 LSP/second. This indicates that LSPs priority 2 are able to acquire resources from priority 3 without being excessively preempted by high priority LSPs when the network is not too highly congested. For operating conditions where the arrival rate is higher than 0.3 LSP/second, the network is significantly biased towards high priority LSPs; in this case, the service obtained by the lower priority LSPs of priorities 2 and 3 (especially priority 3) is very poor and decreases further with increasing load. Therefore, the preemption strategies are best suited for network with medium to high traffic load, beyond which low priority LSPs will be heavily penalized.

In the previous sections, four parameters i.e. τ , σ , ω_1 and ω_2 are proposed for the preemption schemes. The parameters may be set independently based on different objective and operating conditions. The parameter τ can be set with integer values ≥ 0 such that LIP will limit the path search and trigger preemption if necessary. Simulation results show that τ with value 1 or 2 for higher priority LSPs would produce higher throughput without causing excessive preemption events. A generally reasonable value for σ is such that the maximum length of the alternative path used for re-routing is approximately double the length of the average shortest paths in the network. A reasonable candidate for ω_1 and ω_2 is 1 and 10, respectively. However, ω_2 can be adjusted upward or downward to reflect a bigger or simpler network topology.

4.4 Decentralized Preemption Strategy

The preemption strategies presented in Section 4.2 have some drawbacks in real implementations. The first problem is its global re-routing approach, in which a grace period must be allocated to re-route the preempted LSPs one-by-one. The re-routing process involves sending *re-route request* signal to the source node, finding an alternative path, establishing and switching traffic to the alternative path, and terminating original LSPs. The overall cost will be k times of the re-routing cost if there are k existing LSPs to be re-routed. The overhead cost is compounded by the scenario where multiple links are involved in the re-routing process. Apart from the re-routing cost, the new LSP has to tolerate a greater set up delay as well. Secondly, the preemption algorithm requires that every link keep tracks of the paths taken by the LSPs so that the hop count and its congestion level can be properly evaluated. These challenges point to the requirements of a more decentralized approach, such that preemption and re-routing is managed by the link locally. This section attempts to reformulate the solutions by incorporating local re-routing and simple preemption algorithms which nevertheless provide results comparable to the earlier strategies with global rerouting. Local re-route will significantly reduce the overhead cost and delay incurred as alternative paths are only needed to be set up to bypass the preemption link.

The links are segregated into four categories which will be important for the routing algorithm.

- *Admissible Link* – this link is ready to accept the new LSP with its residual bandwidth $U_{ij} \geq b_{new}$.

- *Preemptable Link* – this link can only accept the new LSP by triggering preemption, $U_{ij} + T_{ij} \geq b_{new}$. In addition, all the preempted LSPs can be locally re-routed such that the network throughput is unaffected.
- *Disruptive Link* – this link differs from preemptable link in that not all preempted LSPs can be locally re-routed. Some of the LSPs have to be terminated which will affect the network throughput.
- *Infeasible Link* – this link cannot accommodate the new LSP as $U_{ij} + T_{ij} < b_{new}$.

By defining the links in these four categories, the routing algorithm can search for a path that consists of purely admissible links or with a mix of preemptable and disruptive links. From the viewpoint of network throughput, only admissible link and preemptable link are favorable because existing LSPs are not terminated. However, preemptable link is associated with mandatory local re-routing and thus causes higher overhead cost. Its must be strictly limited to the condition when its selection is beneficial.

Fig. 4.15: Reduction of Overall Traffic Load by Local-Rerouting.

Fig. 4.15 shows how local re-routing may in fact reduce the overall traffic load. All the links in the topology have bandwidth equal to 50 units. LSP 1 with bandwidth 10 is originally routed on the path $G-F$. LSP 2 with bandwidth 50 and source-destination (A, E) arrives later and finds that there are two paths with purely admissible links available, i.e. $A-B-C-D-E$ and $A-G-C-F-E$. By choosing either one of the paths, the total bandwidth consumption of LSP 1 and LSP 2 is 210 units. However, if preemptable link $G-F$ is chosen by re-routing LSP 1, the total bandwidth consumption is 170 units. This choice will therefore provide benefits of overall network load minimization and better future admission success. Following this strategy, routing algorithm may be designed in a way such that the path that minimizes overall network load is selected. Unlike traditional preemption approaches in which preemption is triggered only if path with pure admissible links are not available, one would seek to proactively manage the network resources by taking the advantage of preemption. Although this active management may introduce more re-routing events, simulation results show that the network achieves better performance.

Let us denote the two end nodes of the link that triggers preemption as n_s and n_t , and x_{ij} as the network flow from node i to node j . A maximum flow problem [90] can be constructed to find out the amount of bandwidth that can be re-routed to bypass the link that triggers preemption.

4.4.1 Problem 4.1: Maximum Flow for Local Re-route

Maximize m

Subject to:

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = \begin{cases} m & \text{for } i = n_s, \\ 0 & \text{for all } i \in N - \{n_s, n_t\}, \\ -m & \text{for } i = n_t. \end{cases} \quad (4-7)$$

$$0 \leq x_{ij} \leq U_{ij} \quad \text{for all } E_{ij} \in E \quad (4-8)$$

$$m \leq b_{new} - U_{n_s, n_t} \quad (4-9)$$

The first and third constraints in equation (4-7) define the maximum flows emanating from n_s and end at n_t . The second constraint equation (4-7) is for flow conservation. Constraint (4-8) specifies that the flow must be positive and smaller than the residual bandwidth. Constraint (4-9) is used to limit the maximum flow to the amount of bandwidth that needs to be preempted. By solving the above problem, a set of links with its associated flows that can be used to locally re-route the preempted LSPs will be obtained. For a preemptable link, the maximum flow is $m = b_{new} - U_{n_s, n_t}$. Given the solution of Problem 4.1, the net extra traffic load created due to the local re-route is given by,

$$\Omega_{n_s, n_t} = \sum_{i \in N} \sum_{j \in N} x_{ij} - m \quad (4-10)$$

Since disruptive link is not able to re-route all the LSPs preempted, the solution will give the set of flows with the maximum re-routable bandwidth $m < b_{new} - U_{n_s, n_t}$.

The amount of traffic load created due to the local re-route is also given by (4-10).

The total bandwidth that has to be terminated is,

$$V_{n_s, n_t} = U_{n_s, n_t} - m \quad (4-11)$$

With the set of flows given by the solution in Problem 4.1, the network will know how to divert the traffic of ongoing LSPs so that sufficient residual bandwidth can be reserved for the new LSP. In order to divert the traffic, the MPLS network can set up tunnels on the links with positive flows. This process remains a local activity without the need for source node participation. In MPLS architecture, the node n_s only needs to attach an extra label to the arriving packets before dispatching them to the next hop. This label helps the packets to route through the tunnel and is removed at n_t . Upon the completion of service of the LSPs using the tunnels, it will be torn down accordingly. The whole process is independent of source node and end users.

The attributes of the links are captured with respect to the four categories defined above by using the following variable, θ_{ij} .

$$\theta_{ij} = \begin{cases} b_{new} & \text{if } E_{ij} \text{ is admissible ,} \\ b_{new} + \Omega_{ij} & \text{if } E_{ij} \text{ is preemptable ,} \\ b_{new} + \Omega_{ij} + \rho \cdot V_{ij} & \text{if } E_{ij} \text{ is disruptive ,} \\ \infty & \text{otherwise .} \end{cases} \quad (4-12)$$

Equation (4-12) is specifically designed to guide the routing algorithm in choosing the route favorable for the new request. The scaling factor $\rho > 0$, is used to magnify the bandwidth that needs to be preempted. As the termination of on-going

LSPs will interrupt network services, the use of disruptive links should be minimized. The author proposes that ρ should be sufficiently large to distinguish it from the preemptable links. For the purpose of this study, the value is set to $\rho = CAP_{ij}$. The routing problem is defined as follow,

4.4.2 Problem 4.2: Routing to Minimize the Network Load

$$\text{Minimize } \sum_{(i,j) \in N} \theta_{ij} x_{ij}$$

Subject to:

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = \begin{cases} 1 & \text{for } i = s, \\ 0 & \text{for } i \in N - \{s, t\}, \\ -1 & \text{for } i = t. \end{cases} \quad (4-13)$$

$$x_{ij} \geq 0 \text{ for all } E_{ij} \in E. \quad (4-14)$$

The source and destination of the new LSP are represented by s and t respectively. Constraint (4-13) is used for flow conservation. The routing problem defined in Problem 4.2 is able to find the path that minimizes the overall network load. By minimizing the variable θ_{ij} , the path will include preemptable links only if the total network load including local re-route is lower than the path of purely admissible links. The routing algorithm will avoid disruptive links as far as possible since ρ will make θ_{ij} exceedingly large in comparison to admissible links and preemptable links. Disruptive links will be selected only under the condition that its absence will result in a disconnected graph for the source and destination. Furthermore, ρ serves the purpose of differentiating the various disruptive links.

Disruptive links that terminate less number of existing LSPs are more favorable than those that terminate more LSPs. As a result, service interruption is minimized concurrently.

In order to simplify the preemption algorithm, the score function in (4-6) is replaced with function (4-15) for K existing LSPs on the edge E_{ij} . Every preemptable connection is evaluated using (4-15) and preemption is triggered with the ascending order of (4-15),

$$F_s(c_k) = (b_{new} - U_{ij} - b_k)^2 \quad (4-15)$$

This changes only affect the LSPs c_k at the priority level q which $S_q \geq b_{new} - U_{ij}$. Equation (4-15) is designed such that the LSP that has the closest bandwidth in comparison to the bandwidth that needs to be preempted is selected. Therefore, the number of LSPs and bandwidth preempted will be reduced. Under this preemption strategy, the link only needs to keep track of the residual bandwidth U_{ij} , the LSPs information i.e. bandwidth and priority, and the maximum re-routable bandwidth. This significantly reduces the amount of information needed in Section 4.2 and minimizes the overall signaling overhead.

The path chosen by the routing algorithm may consist of more than one preemptable links and disruptive links. Consequently, the different set of flows given by the solution of Problem 4.1 on different links may interfere. In order to resolve this problem, multi-commodity problem [90] can be used by assuming each of the links that triggers preemption as a single commodity. Let $G' = (N', E')$ denotes the subgraph that eliminates all the edges of the path used by the new LSP,

$E' = E' \cup E_{ij} \notin \{E_{sn_1}, E_{n_1n_2}, \dots, E_{n_m t}\}$. The multi-commodity problem is defined as follows,

4.4.3 Problem 4.3: Multiple Re-routes

$$\text{Maximize } \sum_{1 \leq a \leq A} m^{(a)}$$

Subject to:

$$\sum_{j \in N'} x_{ij}^{(a)} - \sum_{j \in N'} x_{ji}^{(a)} = \begin{cases} m^{(a)} & \text{for } i = n_s^{(a)}, a = 1, 2, \dots, A, \\ 0 & \text{for } i \in N' - \{n_s^{(a)}, n_t^{(a)}\}, a = 1, 2, \dots, A, \\ -m^{(a)} & \text{for } i = n_t^{(a)}, a = 1, 2, \dots, A. \end{cases} \quad (4-16)$$

$$\sum_{1 \leq a \leq A} x_{ij}^{(a)} \leq U_{ij} \quad \text{for all } E_{ij} \in E', \quad (4-17)$$

$$x_{ij}^{(a)} \geq 0 \quad \text{for all } E_{ij} \in E', a = 1, 2, \dots, A, \quad (4-18)$$

$$m^{(a)} \leq b_{new} - U_{n_s n_t}^{(a)}, a = 1, 2, \dots, A. \quad (4-19)$$

Each commodity is denoted by a , which represents the preemptable link or disruptive link. For example, if there are two preemptable links $A = 2$, then the set of flows $x_{ij}^{(1)}$ is used to re-route the LSPs on the first preemptable link and $x_{ij}^{(2)}$ for the second link. Constraint (4-17) ensures that the total flows on a single link will not exceed the residual bandwidth. After obtaining the solution for Problem 4.3, the network will know the amount of bandwidth to be re-routed, how to re-route and trigger preemption accordingly before admitting the new LSP. The flow diagram of the decentralized strategy is illustrated in Fig. 4.16.

Fig. 4.16: The Flow Diagram of the Decentralized Strategy.

4.5 Performance Comparisons

This section compares the performance of decentralized preemption strategy against LIP and SEP using simulations written in C++. The same network topology and simulation parameters here are used as in Section 4.3, i.e. $\sigma = 3$, $\omega_1 = 1$ and $\omega_2 = 10$. In this section LIP refers to Case B as indicated in Section 4.3. For the decentralized preemption strategy, Problems 4.1, 4.2 and 4.3 cited above are solved by using the CPLEX optimizer.

Fig. 4.17: Throughput Comparison of Various Preemption Strategies.

Fig. 4.17 presents throughput performance of different preemption strategies at increasing traffic load. The preemption strategy PREM [81] has low throughputs because it does not incorporate re-routing. Fig. 4.17 also indicates that decentralized preemption strategy has similar performance to SEP. This is due to the fact that preemptable links do not exist in large quantity and thus the network has to use longer paths for most of the cases. Nevertheless, the decentralized strategy shows its effectiveness in improving the throughput as compared to a non re-routing preemption strategy like PREM.

The higher throughput of decentralized strategy as compared to SEP arises because of its higher re-routing probability as shown in Fig. 4.18. This result reflects the important role of re-routing in throughput improvement. As the decentralized preemption strategy is more active in the network resource management than SEP, it

is in a better position to accept more future LSPs. Although re-routing events may increase as a result of this approach, better throughput is attained.

Fig. 4.18: Re-routing Probability of SEP, LIP and Decentralized Strategies.

The preemption probability shown in Fig. 4.19 represents the probability that a new LSP arrival will trigger preemption. It is the ratio of the number of preemptions triggered to the number of LSPs admitted. This ratio measures the relative intensity of the various preemption strategies in executing resource management. As expected, LIP shows the highest preemption probability because high priority LSPs are constantly trying to route on the shortest path. SEP has the lowest preemption probability because, in this case, the network tries to avoid preemption as far as possible. The decentralized strategy participates more actively in the resource management as the network can choose preemptable links in order to minimize overall network load.

Fig. 4.19: Preemption Probability for SEP, LIP and Decentralized Strategies.

Fig. 4.20: Average Path Length for No Preemption, SEP, LIP and Decentralized Strategies.

Fig 4.20 shows the average path length for different strategies. Notice that the decentralized strategy is able to reduce the path length as compared to the system without preemption. This is mainly contributed by preemptable links in which the network can choose the shorter path if the overall traffic load offered is reduced. A check on the LSPs distribution shows that all the schemes have similar performance as illustrated in Fig 4.14.

Given the three preemption strategies proposed, LIP provides the best performance but that it comes with extensive network resource management as the network triggers preemption and re-routing events greedily. This may introduce very large amounts of control overhead. The decentralized strategy shows a balance between LIP and SEP. Its advantages are lower control overhead and less complexity. The network link only needs to keep track of its residual bandwidth and some simple LSP information such as priority and bandwidth. The primary reason that LIP performs better than the decentralized strategy is the effectiveness of global re-routing. Global re-route is able to find a better alternative path because it is source routed, in which the alternative path may be shorter than the original path. Furthermore, LSPs originating from different source-destination pairs have a better chance to be re-routed through global re-routing. In comparison, local re-route will constantly introduce more hops than the original path as the preempted LSPs can only take the routes that bypass the preemption link.

The Problem 4.3 defined in Section 4.4.3 introduces extra computational complexities to the decentralized strategy. However, in most of the cases, fewer than 15% of the LSPs admitted use more than one preemptable link or disruptive link. Therefore, Problem 4.3 can be omitted in resolving the decentralized preemption

strategy. Local re-routing is triggered on the preemptable links or disruptive links one at a time. Some of the existing LSPs will be terminated if there are insufficient resources to route all the to-be-preempted LSPs. The network throughput achieved by omitting Problem 4.3 shows approximately 5% degradation than the results presented in Fig. 4.17. This indicates that the decentralized strategy can attain easier implementation without severe degradation of network performance.

4.6 Summary

In this chapter, three preemption schemes with re-routing mechanisms are presented. The SEP scheme allows LSPs to search for all possible paths before triggering preemption to acquire network resources. The second scheme, LIP, seeks to limit the search space so that new high priority LSPs are assigned with shorter paths. In both schemes, global re-routing mechanisms are used to route the to-be-preempted LSPs on alternative paths so that network throughput will not be adversely affected. The studies show that networks achieve better throughput by implementing LIP. This is due to the fact that by routing the high priority LSPs on shorter paths, the overall network resource consumption is minimized and thus more future LSPs can be admitted. SEP shows poorer performance because the high priority LSPs that search all the possible paths may at times use relatively longer paths. For ease of deployment, a decentralized preemption scheme with local re-routing is formulated which gives comparable results to SEP and LIP. The decentralized scheme uses routing algorithm to find the path that consumes minimum network resources. In order to minimize the loss of throughput, links that can locally re-route the existing LSPs are selected ahead of the links that have to terminate most of the existing LSPs.

The results indicate that by more actively managing the network resources, we can satisfy the requirements of higher priority LSPs and achieve better overall performance at the same time.

The implementation of any one of the schemes is application specific. For application that needs strict control on path length and delay, LIP provides the best result but with the cost of higher control overhead. Although the decentralized strategy is not able to perform as well as LIP, it involves less control overhead and complexity. Therefore, decentralized strategy is more suitable for applications that do not have high requirements.

5. Stochastic Petri Net Model for Preemption

5.1 Motivation

The primary motivation for this chapter is the lack of appropriate analytical tool for the performance of network preemption system. To the best of our knowledge, this is the first attempt at modeling network preemption using Stochastic Petri Nets (SPN). SPN is a powerful modeling tool and the basic structure underlying a SPN is a Markov chain [94], [96]. The generic SPN model is designed such that it can provide insights into the performances of different preemption strategies. It can also serve as the platform to compare any new preemption algorithms.

In this chapter, the author firstly proposes a single link preemption model. It takes into consideration the different priority levels and bandwidth requested. The model can be used to analyze a single link that supports K multiple connections. This single link model is then generalized and applied to networks. Without loss of generality, a fixed routing approach is used in the network model where paths are pre-computed for every node pair. This will reduce the complexity of the SPN model but is sufficient to reveal the network performance. Network connections are arriving based on the Poisson process and randomly assigned with different priority levels, bandwidth, and source-destination node pairs.

5.2 Brief Description on Stochastic Petri Net

For completeness, a short description of SPN is given here. The basic structure underlying a SPN is a Markov chain. A SPN is constructed by two types of nodes – places and transitions. Places are shown as circles in the net, which are used to represent conditions. Places can contain zero or a number of tokens to distinguish different conditions. Tokens are drawn as black dots. Transitions represent occurrence of events. Two types of transitions can be used – timed-transitions and immediate transitions. Timed-transitions will fire (enable) after a delay which is determined by the distribution specified for it. Time-transitions are shown as unfilled rectangles. Immediate transitions, shown as bars, will fire immediately when enabled. Fig. 5.1 shows these basic elements of SPN. If a timed-transition and an immediate transition are enabled concurrently, the immediate transition has the priority to fire.

Fig. 5.1. Basic Elements of SPN.

Arcs, represented by arrows, are used to connect places to transition (input arc) and transitions to places (output arc). When a transition fires, it will draw an amount of tokens, equals to the cardinality of the input arcs, from its input places. Likewise, the transition will deposit an amount of tokens, equals to the cardinality of the output arcs, into its output places. A transition is enabled when each input place has as many

or greater number of tokens than the input arc cardinality. On the contrary, inhibitor arc is used to disable transitions when the input place has as many or greater number of tokens than the inhibitor arc cardinality.

5.2.1 Stochastic Petri Net Package (SPNP) Modeling Tool

Fig. 5.2: SPNP Modeling Tool.

One of the well designed tools for analyzing SPN model is the SPNP [89] modeling tool. The basic language used in SPNP is C-based SPN Language (CSPL), which is an extension of C programming language. However, with the latest released package that features intuitive Graphical User Interface (GUI), the need for users to construct the SPN model using CSPL is greatly reduced. SPNP takes input models from users and solves it numerically through its algorithms. Various key performance results such as transient and steady-state can be computed. A reachability graph will be

constructed by SPNP if the model is solved analytic-numeric method. The GUI for SPNP is illustrated in Fig. 5.2.

5.3 SERA Preemption Algorithm

One of the most important performance metrics in preemption is throughput. Throughput is the dot product of bandwidth used and total service time and it is only accrued when a connection completes successfully. It reflects the effectiveness of a preemption scheme in utilizing the network resources. However, instead of using connection throughput directly, most of the existing preemption schemes [80-87] use proxy criteria such as bandwidth and number of connections to preempt connections. The primary reason remains that the connection service time is not available to the network in advance. Hence, the author attempts to develop a Service Ratio Arbitrated (SERA) preemption strategy that preempts connections using estimated connections service time information. This preemption strategy basically tries to preempt the connection that has just started in order to minimize the loss of throughput as compared to a non-preemption link. Next, performance evaluation is conducted to compare SERA with a priority-based strategy using the SPN model developed.

5.3.1 Formulation of SERA

Let us assume that a connection k has a holding time t_k^h with starting and ending times denoted as t_k^s and t_k^d . Assuming that the link's capacity as CAP and that a new connection, c_{new} , arrives with bandwidth, b_{new} , at time t as shown in Fig. 5.3. This will find the link congested if the idle bandwidth at time t is less than the b_{new} required.

Fig. 5.3: Bandwidth and Holding Time of Connections.

The objective of the preemption strategy is to maximize the network throughput. For this connection k , its throughput η_k , is defined by the product of connection holding time and its bandwidth,

$$\eta_k = b_k \cdot t_k^h \quad (5-1)$$

Let us assume that the throughput of the preempted connection is not accrued. This can happen if the connection is meant for business transaction, telephone call, streaming video and other services that are connection-oriented. Once interrupted, the connections may have to be restarted. Although the straightforward strategy may be to preempt the connection with the lowest throughput, Bar-Noy *et al.* [76] proved that this will not lead to the optimization of throughput. Instead, they stated that only the connection holding time is sufficient to design a feasible solution. From Fig. 5.3, we can derive the connection elapsed time as $t_k^e = t - t_k^s$ and the remaining holding time

as $t_k^r = t_k^d - t$. We define the connection service ratio, $\delta_k(t)$ as the ratio of the remaining holding time to the elapsed time plus the connection start time,

$$\delta_k(t) = \omega_1 t_k^s + \omega_2 \frac{t_k^r}{t_k^e}, \quad t_k^e > 0 \quad (5-2)$$

where $\omega_1 \geq 0$ and $\omega_2 \geq 0$ are the weights attached to (5-2). The service ratio represents the relative progression of the existing connections at time of observation t . Since throughput is gained only after the completion of service, the motivation is to preserve the connections that are near completion and preempt the ones that have just started. Since the elapsed time, t_k^e , is monotonically increasing and the remaining time, t_k^r , is monotonically decreasing, the service ratio, $\delta_k(t)$, is a monotonically decreasing function. Therefore, the connection with the highest $\delta_k(t)$ should be preempted first in order to achieve the objective of preserving the connections that near completion. The proposed strategy is referred to as *Service Ratio Arbitrated* (SERA) preemption strategy.

Equation (5-2) requires the connection duration information as *a priori* to derive the remaining holding time. This may be difficult to obtain in practice as typical calls do not pre-announce (or even know) their holding times in advance. Instead, one would seek to approximate the connection holding time based on the statistical information such as the mean holding time and its variance that may be available from historical data or prior measurements. Given the holding time distribution, the (cumulative distribution function) CDF and (probability density function) pdf of the remaining holding time are (5-3) and (5-4), respectively.

$$\begin{aligned}
 F_{t_k^r}(t) &= P\{t_k^r \leq t \mid t_k^h > t_k^e\} = P\{t_k^h \leq t + t_k^e \mid t_k^h > t_k^e\} \\
 &= \frac{F_{t_k^h}(t + t_k^e) - F_{t_k^h}(t_k^e)}{1 - F_{t_k^h}(t_k^e)}, \quad t \geq 0
 \end{aligned} \tag{5-3}$$

$$f_{t_k^r}(t) = \frac{dF_{t_k^r}(t)}{dt} = \frac{f_{t_k^h}(t + t_k^e)}{1 - F_{t_k^h}(t_k^e)}, \quad t \geq 0 \tag{5-4}$$

The remaining mean holding time is then given by,

$$\begin{aligned}
 \overline{t_k^r} &= \int_0^\infty \frac{t}{1 - F_{t_k^h}(t_k^e)} f_{t_k^h}(t + t_k^e) dt \\
 \overline{t_k^r} &= \frac{\overline{t_k^h} - t_k^e - \int_0^{t_k^e} (t - t_k^e) f_{t_k^h}(t) dt}{1 - F_{t_k^h}(t_k^e)}
 \end{aligned} \tag{5-5}$$

Note that (5-5) requires us to know the probability distribution model of the connection holding time in order to obtain the mean remaining holding time. However, this distribution model may not be available. As such, an approximation to the probability distribution model, $\hat{F}_{t_k^h}(t)$ is used which depends on the squared coefficient of variation c_T^2 of $F_{t_k^h}(t)$, where $c_T \equiv \sigma_k / \overline{t_k^h}$, $\overline{t_k^h}$ is mean holding time and σ_k is standard deviation. For $0 < c_T^2 \leq 1$, one can fit an $E_{n-1,n}$ distribution and for $\frac{1}{2} \leq c_T^2$ one can fit a Coxian-2 distribution [95]. The definition of the $E_{n-1,n}$ and Coxian-2 distribution is provided in Appendix 1 along with the method [95] to choose the parameters to get a 2-moment fit. Thereby, equation (5-2) is rewritten as,

$$\delta_k(t) = \omega_1 t_k^s + \omega_2 \frac{\overline{t_k^r}}{t_k^e}, \quad t_k^e > 0 \tag{5-6}$$

Using equation (5-6), a link that triggers preemption can evaluate the individual existing connections based on the service ratio, $\delta_k(t)$, and arrange the connections in the descending order of $\delta_k(t)$. In order to preserve the connections that near completion, the one with the highest $\delta_k(t)$ is preempted first until the bandwidth requested by the new connection is satisfied. To simplify the approach, the subsequent SPN model only focuses on minimizing the loss of the network throughput without differentiating the priority levels of the connections that may be preempted. Since connections of higher priority may carry higher economical values, SERA may be modified to take into consideration the priority levels. For example, the link may preempt connections from the lowest priority until priority p such that the sum of bandwidth from connections of priority p exceeds the bandwidth requirement. Under such condition, the service ratio $\delta_k(t)$ can be used to select the combination of connections (priority p) for preemption.

5.3.2 Properties of SERA

This section examines the behavior of the SERA preemption algorithm in three special cases depending on the sequence of connections that have arrived. Let C_s be the sequence of K connections, $C_s = c_1, c_2, \dots, c_K$. The author would like to investigate the performance of the algorithm for a subsequence, $C'_s \subseteq C_s$, where $C'_s = c_1, c_2, \dots, c_g$, as in Fig. 5.4, that triggers preemption. This indicates that the sum of bandwidth from connections in the sequence C'_s exceeds the link capacity. Since connections arrive one at a time, the link will find the start time of every subsequent connection is later than the previous connections, i.e. $t_1^s < t_2^s < \dots < t_{g-1}^s$. It is also assumed that the last connection in the sequence C'_s , i.e. c_g , will trigger preemption.

Fig. 5.4: Subsequence of Arriving Connections.

Case 1: Assuming that connections in the sequence C'_s can be ordered such that the end time of every subsequent connection is later than its previous connections, and that the holding time of every subsequent connection is shorter than its previous connection, i.e. $t_1^d \leq t_2^d \leq \dots \leq t_{g-1}^d$ and $t_1^h > t_2^h > \dots > t_{g-1}^h$. When the last connection, $c_{g'}$, arrives at time $t_{g'}^s$, the order of the service ratio is given by $\delta_1 < \delta_2 < \dots < \delta_{g-1}$. Intuitively, if there are only two existing connections and given the arrival time of the connection $c_{g'}$ is at $t_{g'}^s$, the remaining service time of the first connection in the sequence C'_s is shorter than the remaining time of the second connection, i.e. $t_1^d - t_{g'}^s < t_2^d - t_{g'}^s$. Also, the elapsed time of the first connection is larger than the elapsed time of the second connection in the sequence, i.e. $t_{g'}^s - t_1^s > t_{g'}^s - t_2^s$. Applying the observations into (5-2) gives $\delta_1 < \delta_2$. The result can be generalized to prove the claim above. Essentially, SERA preemption algorithm will remove the connections

from the bottom of the sequence C'_s . One may argue that this decision is as good as the offline strategy because by preempting from the bottom, the longer connections that are about to finish tend to be preserved and thereby minimizing the throughput degradation.

Case 2: This case is similar to case 1 in the ordering of connections. All the conditions are similar as well except that the holding time of the subsequent connections in the sequence C'_s is longer than previous connections, i.e. $t_1^h < t_2^h < \dots < t_{g-1}^h$. By using the same method as in case 1, the service ratios of the existing connections are given by $\delta_1 < \delta_2 < \dots < \delta_{g-1}$. The proposed SERA algorithm will preempt the connection with the largest δ although it has the longest holding time. This is counter intuitive because a longer connection may give higher throughput. If we assume that the system stops after the arrival of the new connection c_g , an offline preemption strategy will preempt the connection with the smallest δ to minimize the loss of throughput. Therefore, discrepancies exist between the proposed strategy and a perfect offline strategy. As SERA tries to take the advantage of improving the throughput quickly by preserving connections that are going to complete, it risks losing potential higher throughputs from longer connections. However, in real implementations when connections arrival sequence is not available in advance, a longer connection that is far from completion may suffer from a higher probability of being eventually preempted. Thus, the proposed strategy may perform better at higher connection arrival rates.

Let $\eta(C'_s)$ be the throughput preempted by the offline strategy that starts from the top of sequence C'_s and $\eta_p(C'_s)$ as the throughput preempted by SERA. Let χ be the ratio of $\eta(C'_s)$ to $\eta_p(C'_s)$, which is used to compare the throughput preempted

from both strategies. With a higher value of χ , the performance of the proposed strategy on sequence C'_s is closer to the offline strategy. The worst performance by SERA arises when only one connection is preempted.

$$\chi = \frac{t_{1'}^h \cdot b_{1'}^h}{t_{g-1'}^h \cdot b_{g-1'}^h} \quad (5-7)$$

The numerator in (5-7) represents the throughput of the first connection in sequence C'_s whereas the denominator represents last connection. Recall that an offline strategy will preempt the connection with the smallest δ (i.e. first connection in sequence C'_s) in order to maximize total throughput. Let u_t be the ratio between the largest and smallest values of connection's holding time and u_b be the ratio between the largest and smallest values of bandwidth. The lower bound performance of SERA to an offline strategy is given by $\chi \geq (u_t u_b)^{-1}$.

Case 3: In this case, the connections are assumed to arrive in such a way that the end time of every subsequent connection is earlier than previous connections, i.e. $t_{1'}^d > t_{2'}^d > \dots > t_{g-1'}^d$. Assume that there are only two existing connections in the sequence, i.e. $c_{1'}$ and $c_{2'}$, and let $\omega_1 = 0$, $\omega_2 > 0$. The exact arrival time, t , of the new connection $c_{g'}$ where SERA is indifferent between the two existing connections is given by,

$$t = \frac{t_{1'}^d t_{2'}^s - t_{2'}^d t_{1'}^s}{t_{1'}^h - t_{2'}^h} \quad (5-8)$$

This implies that for two existing connections, if the arrival of c_g is earlier than t , c_2 will be preempted. Otherwise, c_1 is preempted if c_g arrives later than t . Since c_1 can potentially provide more throughput as it has a longer service time, the network can adjust the weighting factor ω_1 in (5-6) to have higher value so that c_2 is preempted. The lower bound performance of this case is also the same as in case 2. Generally, the weighting factor ω_1 will not affect the preemption decision in case 1 and case 2. However, for case 3 one can assign the weighting factor such that $\omega_1 \geq \omega_2$ in order to preempt the connection with shorter holding time.

5.3.3 Non-Preemption SPN Model

Fig. 5.5: Non-Preemption SPN Model.

In this section, a new SPN preemption model for a single link case is presented. Firstly, the author develops a model that excludes preemption. It uses the number of tokens to keep track the priority levels assigned as well as the bandwidth allocated. Connections arrived that find sufficient resources at the link will be admitted immediately. Otherwise, it will be rejected if the link is congested. Since preemption

decisions need explicit existing connection information, the model is designed such that this information is explicitly represented. Although this will inevitably increase the complexity of the model, important results can be measured for a model with sufficient solvable size.

The model of a single link without preemption that supports two priority levels and two bandwidth levels is shown in Fig. 5.5. The timed transition ARR represents the arrival of new connections. Upon arrival, one token will be deposited to place $P0$ and $P1$ respectively. Immediate transitions $t0$ and $t1$ with equal probability will fire to put a number of tokens at $P2$ to represent the bandwidth requested. Whenever transition $t0$ fires, one token will be deposited. Otherwise $t1$ will deposit two tokens. Similarly, $t2$ or $t3$ will deposit one or two tokens to $P3$ to represent the priority level.

The transitions $t4$ and $t5$ represent the connections that can be concurrently on service. The places $P6$ and $P9$ with inhibitor arc to $t4$ and $t5$ respectively, will not admit the new connection if both of them have a token, which represents there are existing connections on service. If $P6$ is empty, $t4$ will fire to destroy the tokens at $P2$ and put the same number of tokens at $P4$. The same amount of tokens will also be drawn from $P12$ that represents the link capacity. Simultaneously, $t4$ will put the priority level at $P5$ and one token at $P6$. Time transition $T1$ and $T2$ represents the connection holding time. When $T1$ fires, the token at $P6$ will be destroyed and one token deposited at $P10$. Immediately, $t6$ will fire to draw the bandwidth and priority level from $P4$ and $P5$ and deposit the same amount of tokens at $P4$ to $P12$ to signify the release of bandwidth.

The immediate transition $t8$, with a guard function specifying that if both $P6$ and $P9$ are not empty or $P12$ has smaller number of tokens than $P2$, will fire to destroy

tokens at $P2$ and $P3$ to represent connection rejection due to insufficient bandwidth. Place $P13$ will have two tokens if $t8$ fires. A new arrival will destroy all the tokens at $P13$ and deposit one back at $P13$. The probability of $P13$ with 2 tokens represents the blocking probability.

5.3.4 Preemption SPN Model

Fig. 5.6 Preemption SPN Model.

In this section, Fig. 5.6 now presents the important modifications to the non-preemption system to allow for preemption. The immediate transition $t9$ has higher priority than $t8$. Therefore, in the event of insufficient bandwidth, the link will check for the possibility of preemption before rejecting the new connection. Transition $t9$ contains a guard function that checks for enough preemptable bandwidth and determine the existing connection to be preempted based on the preemption algorithm. One connection is preempted at a time until the tokens at $P12$ is higher than $P2$. The existing connections are indexed with k , i.e. $k = 1, 2, 3, \dots, K$ and $t9$

will put k number of tokens to $P15$ if connection k is preempted. As in Fig. 5.6, $t10$ will trigger if $P15$ has one token; $t11$ will trigger if $P15$ has two tokens.

If $t10$ fires, the token at $P6$ will bypass $T1$, therefore the connection will release resource immediately and return the number of tokens that represent bandwidth to $P12$. Given a new connection that triggers preemption, $t9$ will deposit 2 tokens at $P13$ for the first connection that is preempted and 0 token subsequently. Therefore, the preemption probability is represented by the probability of $P13$ having 3 tokens.

Table 5.1: Immediate Transitions Firing Priority for Single Link Model.

Firing Priority	Immediate Transitions
4 (highest)	$t4, t5, t6, t7$
3	$t9, t10, t11$
2	$t8$
1	<i>All others</i>

Table 5.1 illustrates the firing priorities for the different immediate transitions used in the models. These priorities ensure the proper firing sequence of immediate transitions so that the preemption system will function appropriately. The models developed above can be generalized to analyze a single link that supports K connections by placing more nodes as similar to $P4, P5, P6, P10, t4, t6, t10, T1$ and connected in the same ways.

5.3.5 Numerical Results

By assuming Poisson arrival process, exponential service time, 3 bandwidth levels, 3 priority levels, link capacity of 8 tokens, and 8 concurrent connections, the sizes of

the Markov chains are given in Table 5.2. For the priority-based strategy, the link will preempt the connections from the lowest priority level until the bandwidth requested is satisfied. The ‘Centralized’ scheme represents the centralized preemption algorithm proposed in Chapter 3. Since the SPN model described in the above sections is designed for a single network link, preempted connections will not be re-routed.

Table 5.1: Sizes of the Markov Chains for the Single Link Model.

Model	No. States	No. Transitions	No. Tangible Markings	No. Non Zero Entries	Time Taken
No Preemption	38	22	1,213	32,667	10 mins
Preemption	39	31	2,354	52,817	10 mins

Fig. 5.7: Link Normalized Throughput.

Fig. 5.7 shows that SERA performs better than the priority-based and ‘Centralized’ strategies. This indicates that the proposed algorithm is better at preventing the loss of throughput by making use of the available connections service

time information. ‘Centralized’ is marginally better than priority-based because it takes into account other criteria such as bandwidth and number of connections. The system without preemption achieves the highest utilization as no existing connection is preempted.

Fig. 5.8 and Fig. 5.9 show the blocking probability and preemption probability respectively. Blocking probability is given by the probability that $P13$ has two tokens whereas preemption probability is given by the probability that $P13$ has three tokens. Preemption probability illustrates the probability that a new connection will trigger preemption. ‘No Preemption’ shows a higher blocking probability as new connections will be rejected regardless of its priority levels if the link is congested. It may be noted that all the preemption schemes perform similarly in terms of blocking probability and preemption probability. Based on the results, SERA can be considered as a better strategy as it produces higher throughput without seriously affecting other performance metrics.

One could see that this SPN model is versatile as it can be used to compare the performance of different preemption strategies. The primary effort needed is to formulate the strategy and implement it at the preemption guard function.

Fig. 5.8: Blocking Probability.

Fig. 5.9: Preemption Probability

5.4 Network Preemption SPN Model

5.4.1 Preemption Strategy

In this section, a simple network preemption strategy based on priority levels is introduced. The preemption strategy described here is primarily used to demonstrate the construction of the SPN model. Re-routing scheme is not implemented as it highly increases the SPN model complexity. The network architecture used in this formulation is based on MPLS networks, as similar to Chapter 4, Section 4.1. Each connection has a priority level ranging over 0–7, where 0 is the highest priority. Let us denote a connection request of priority level p and bandwidth b as $c=(b, p)$. A new connection is denoted by $c_{new}=(b_{new}, p_{new})$. Let l denotes the path used to route the connection.

Let the network graph be $G = (N, E)$, where N is the set of nodes and E is the set of edges. For the link of two adjacent nodes i and j , $E_{ij} \in E$, we denote the maximum link bandwidth capacity as CAP_{ij} and the idle bandwidth as U_{ij} . Let $l_{st} \subseteq E$ be a subset of links that are used to route the connection given the source $s \in N$ and destination $t \in N$. The network is able to admit a new connection if all the links of the path used can satisfy the bandwidth requested, $b_{new} \leq U_{ij}$ for all $E_{ij} \in l_{st}$. However, if one of the links does not have enough idle bandwidth, the new connection can trigger preemption to acquire resources from lower priority connections. Assuming that there are R number of existing connection on link E_{ij} , i.e. $c_1^{ij}, c_2^{ij}, \dots, c_R^{ij}$, the sufficient condition to trigger preemption is given by $b_{new} \leq U_{ij} + T_{ij}$, where T_{ij} is the preemptable bandwidth on E_{ij} .

$$T_{ij} = \sum_{r=1}^R b_r^{ij} I_{\xi}(b_r^{ij}), \quad (5-9)$$

$$I_{\xi}(b_r^{ij}) = \begin{cases} 1 & \text{if } b_r^{ij} \in \xi, \\ 0 & \text{if } b_r^{ij} \notin \xi. \end{cases}$$

$$\xi = \{c_r^{ij} : p_{new} < p_r^{ij}\}.$$

where $I_{\xi}(b_r^{ij})$ is the indicator function for the event ξ that the new connection has higher priority than the existing connections on link E_{ij} . Therefore, T_{ij} represents the sum of bandwidth from existing connections with priority level lower than p_{new} . Next, one would need to determine the combination of existing connections to be preempted in order to accommodate the new connection.

For simplicity and illustration purposes, we use priority-based preemption in which connections are preempted from the lowest priority in order to preserve the integrity of priority level. On link E_{ij} , let S_p be the sum of bandwidth from connections with priority p .

$$S_p = \sum_{r=1}^R b_r^{ij} I_{\phi}(b_r^{ij}), \quad (5-10)$$

$$I_{\phi}(b_r^{ij}) = \begin{cases} 1 & \text{if } b_r^{ij} \in \phi, \\ 0 & \text{if } b_r^{ij} \notin \phi. \end{cases}$$

$$\phi = \{c_r^{ij} : p_r^{ij} = p\}.$$

where $I_{\phi}(b_r^{ij})$ is an indicator function for the event ϕ that the existing connections have the same priority as p . All the connections of priority p will be preempted if $S_p \leq b_{new} - U_{ij}$. This preemption process will start from the lowest priority until the specific priority level q such that $S_q > b_{new} - U_{ij}$. Not all the connections of priority q would need to be preempted if the sum of bandwidth from the existing connections is

higher than that of the new request. In order to minimize the bandwidth preempted, one could use the following function (5-11) to evaluate whether the existing connections are suitable for preemption.

$$F_h(c_r) = [b_r - (b_{new} - U_{ij})]^2 \quad (5-11)$$

For every connection of priority level q is evaluated independently using (5-11) and its value recorded. The connections are then ordered in the descending order of its value given by (5-11). The preemption algorithm will preempt from the top of the list until the bandwidth required is satisfied. Since preemption may happen over multiple links in the network, each of the links will execute preemption independently in order to ensure the scalability of the strategy.

Without loss of generality, let us assume that fixed routing is implemented. Thus, given a source-destination node pair, the network will know in advance which routes (i.e. the collection of links) are used. This will reduce the complexity of the SPN model but is sufficient to reveal the network performance.

5.4.2 Network Preemption Model

The generic SPN model for the network preemption system discussed in Section II is explained in detail in this section. The model is subdivided into three subnets, namely i) *new arrival subnet*, ii) *link subnet* and iii) *connection depository subnet*. Each of the subnets model the following, arrival of new connection, link bandwidth utilization and connection information, respectively. These subnets will interact with each other to emulate the important elements of a network preemption system.

5.4.2.1 New Arrival Subnet

Fig. 5.10: New Arrival Subnet.

Let us assume that a new connection carries a three-tuple information, i.e. bandwidth required, priority level and the source-destination node pair. This information is chosen randomly from some pre-defined values. The new arrival subnet is illustrated in Fig. 5.10. Timed-transition *Arr* represents the arrival of new connections. It will fire after a delay which is exponentially distributed. When *Arr* fires, it will deposit one token into *P0*, *P1* and *P2*, respectively. *P0* is responsible for assigning priority level to a new connection. With one token at *P0*, immediate transitions *p1* and *p2* are enabled. Both *p1* and *p2* are assigned with the same priority and firing priority. Thus, one of them will fire arbitrarily to deposit an amount of tokens into *Prio* to represent the connections priority level. For example in this model, *p1* will fire to deposit one token (represents priority level 1) whereas *p2* will fire to deposit two tokens (represents priority level 2).

Similarly, $b1$ or $b2$ will fire arbitrarily to deposit one token or two tokens respectively into Bw to represent the bandwidth requested by a connection. The immediate transitions $np1$, $np2$, and $np3$ are responsible for selecting the source-destination node pair for the new connection by depositing one token, two tokens and three tokens respectively into $NodePair$. In order to support more priority levels, bandwidths and node pairs, one can place additional immediate transitions in a similar way and assign the arcs with increasing cardinality. The inhibitor arc from $Prio$ to Arr is to stop Arr from firing when the net is processing a new connection. Hence, this also signifies that no two or more connections will arrive at the network concurrently.

The tokens at $Prio$, Bw and $NodePair$ represent the information carried by a new connection. This connection can cause the network to react in three different ways: i) admit the connection, ii) trigger preemption if all the links have sufficient preemptable bandwidth and iii) reject it if some of the links do not have sufficient resources. These three actions are mutually-exclusive and only one of them will be triggered. In Fig. 5.10, immediate transition $Release$ is used to represent the connection admission action. It is associated with an $ADMIT$ guard function which checks for the availability of resources at every link. If the conditions are met, it will fire to remove all the tokens from $Prio$, Bw and $NodePair$. This clearing of tokens allows Arr to fire again (after an exponential delay). The actual storing of connection information is done by connection depository subnet, which will be discussed in the subsequent section.

On the other hand, connection rejection is represented by the immediate transition $Reject$ as shown in Fig. 5.10. Let us define a guard function, $DENY$, at

Reject such that it will fire if some of the links do not have sufficient resources. As similar to *Release*, *Reject* will draw all the tokens from *Prio*, *Bw*, and *NodePair*. However, no connection information will be stored as the connection is rejected. In order for the connection to trigger preemption successfully, it must satisfy $b_{new} \leq U_{ij} + T_{ij}$, for all the links used by the source-destination node pair. Thus, we associate a guard function, *PREEMPT_ALERT*, at *Preempt* with the above condition to function as a flag if preemption can be triggered. When *Preempt* fires, it will deposit one token into *State* and make the sum of tokens at *State* to two. This indicates that preemption is activated and the connection depository subnet will take actions to terminate some existing connections. After an appropriate amount of resources is released, the new connection will be admitted. The inhibitor arc from *State* is to prevent *Preempt* from continuing to fire when the network is terminating some of the existing connections.

Subsequently when *Arr* fires to signify the arrival of another connection, it will draw all the tokens from *State* and deposit one token back. This will maintain one token at *State* if no preemption is activated. The pseudo-codes for *ADMIT*, *DENY*, and *PREEMPT_ALERT* functions are shown in Fig. 5.11, Fig. 5.12 and Fig. 5.13 respectively.

```

Function ADMIT
Given  $Prio, Bw, NodePair$ 
For all the links  $E_{ij} \in l$ 
  If  $Bw > U_{ij}$ 
    Return FALSE
Return TRUE

```

Fig. 5.11: *ADMIT* Guard Function.

```

Function DENY
Given  $Prio, Bw, NodePair$ 
For all the links  $E_{ij} \in l$ 
  If  $Bw \leq U_{ij} + T_{ij}$ 
    Return FALSE
Return TRUE

```

Fig. 5.12: *DENY* Guard Function.

```

Function PREEMPT_ALERT
Given  $Prio, Bw, NodePair$ 
If  $Bw < U_{ij}$  for all  $E_{ij} \in l$ 
  Return FALSE
If  $Bw \leq U_{ij} + T_{ij}$  for all  $E_{ij} \in l$ 
  Return TRUE
Return FALSE

```

Fig. 5.13: *PREEMPT_ALERT* Guard Function.

5.4.2.2 Link Subnet

Fig. 5.14: Link Subnet

The network link is modeled by two places and two immediate transitions, as shown in Fig. 5.14. A network with $|E|$ links needs to have $|E|$ copies of the subnet, in which each copy represents one unique network link. *Cap* with initial markings of C tokens represents the link capacity. If a new connection is accepted, *Draw* will fire to draw an amount of tokens equal to the number of tokens at *Bw* (Fig. 5.10) and put them into *Used*. This reduces the idle bandwidth that can be used by subsequent connections. When the connection is completed or preempted, an equal amount of tokens will be returned to *Cap* through the firing of *Return*. The arc cardinality, $\#b1$, refers to the bandwidth information stored at the connection depository subnet (Fig. 5.17). The guard function defined at *Draw*, $DRAW_BW$, contains the similar function as in *ADMIT* (Fig. 5.11) but with additional condition to verify if the network link it represents is chosen. Similarly, the guard function at *Return*, $RETURN_BW$, will fire if the completed connection uses its resources. Both of these functions are illustrated in Fig. 5.15 and Fig. 5.16.

```

Function DRAW_BW
If ADMIT is TRUE
  If this link is used
    Return TRUE
  Return FALSE

```

Fig. 5.15: *DRAW_BW* Guard Function.

```

Function RETURN_BW
If connection A is completed
  If this link is used
    Return TRUE
  Return FALSE

```

Fig. 5.16: *RETURN_BW* Guard Function

5.4.2.3 Connection Depository Subnet

Fig. 5.17: Connection Depository Subnet.

The connection depository subnet is shown in Fig. 5.17. It consists of M groups of *Depository* and a copy of *Link check*, where M is the maximum number of

connections that can be supported by the network. M is upper bounded by $(|E| * C / b_{\min})$, where C is the link capacity and $b_{\min}$ is the minimum bandwidth requested by a connection. The depository is responsible for keeping the connection specific information such that it can be used to evaluate preemption decision. The immediate transition *Admit* is associated with *ADMIT* guard function and it will fire when a connection can be admitted. It will deposit one token to $P4$, which in turn enables immediate transitions ($D1, D2, \dots, DM$) that have not fired and contain previous connection information. For instance, if $D1$ has fired previously and is currently still holding the connection information, it cannot fire again until the connection is completed. In other words, subsequent connections admitted will only enable $D2$ to DM . Let us assign $D1$ to DM with equal firing priority so that one of them will fire arbitrarily.

Assuming that $D1$ has fired, it will put one token to $s1$ and the same number of tokens from $Bw, Prio$ and $NodePair$ to $b1, p1$ and $n1$ respectively. The token at $s1$ will enable *Server* and it will fire after a delay which represents the connection service time. The service time is assumed to be exponentially distributed. When *Server* fires (connection is completed), one token will be put into $P5$ and immediately *Depart* will fire. This represents the completion of connection and all the tokens from $b1, p1$ and $n1$ will be removed. Recall that $\#b1$ tokens will be returned from *Used* to *Cap* (Fig. 5.14) for the links that have been used by the departing connection.

The subnet denoted by *Link check* in Fig. 5.17 is used to indicate which of the links should trigger preemption. Assuming that the preemption condition is met, *Preempt_Link* (associated with *PREEMPT_ALERT* function) will fire. The output arc connecting *Preempt_Link* to *Link_id* will deposit an amount of tokens equals to the

network link identification into *Link_id*. This mechanism is intended to check for preemption at all the links to be used by the new connection. It will proceed from the source link to the destination and only the links that need to trigger preemption will be identified. As such, existing connections that utilize the identified link will be evaluated for preemption. The actual preemption decision is made by *Bypass*. Its guard function, *REAL_PREEMPT*, will evaluate the connections based on the pre-defined strategy and one or more connections will be terminated until the bandwidth requirement is satisfied. Once a link has freed enough bandwidth, *Clear* will fire to draw all the tokens from *Link_id*. Next, *Preempt_Link* will fire if necessary to trigger preemption on the subsequent link.

If a connection is preempted, *Bypass* will fire and cause the connection information being cleared from the depository immediately. This is exactly the same as if the connection is completed at that instance. When the preemption process is finished, all the immediate transitions tasked with triggering preemption will be disabled as the necessary conditions for preemption are no longer valid. In contrast, the network will find that it is able to admit the new connection and the admission process is activated thereafter. Thus, these subnets complete the task of connection admission, rejection, and preemption. The guard function for *REAL_PREEMPT* is illustrated in Fig. 5.18.

```

Function REAL_PREEMPT
Given Bw, Prio, NodePair
  If link identified is used
    If Prio is among the lowest
      If sum Bw (at Prio) < requested
        Return TRUE
      Else
        Rank current connection according to (5-11)
        If top of the list
          Return TRUE
    Return FALSE

```

Fig. 5.18: *REAL_PREEMPT* Guard Function.

The *REAL_PREEMPT* guard function defined in Fig. 5.18 is implementation specific. Here, we define it in accordance to the general preemption strategy as discussed in Section 5.4.1. In order to ensure the correct firing sequence of all the immediate transitions, its associated firing priorities are as shown in Table 5.3. Immediate transitions that are tasked with connection admission are generally given the highest priority. This is followed by connection preemption and rejection.

Table 5.3: Immediate Transitions Firing Priority for Network Model.

Firing Priority	Immediate Transitions
4 (highest)	<i>Admit, Draw, Return, DI, ... , DM, Depart</i>
3	<i>Release, Preempt, Preempt_Link, Clear, Bypass</i>
2	<i>Reject</i>
1	<i>All others</i>

5.4.3 Numerical Results

Based on the SPN model discussed above, one can derive two main performance measures of a preemption system, namely the network throughput and preemption

probability. Preemption probability is defined as the probability that a new connection will trigger preemption. It is given by the probability that *State* (Fig. 5.10) has two tokens. On the other hand, network throughput is given by the sum of throughput at all the *Server* transitions (Fig. 5.17). As preempted connections will bypass *Server*, the network throughput measured by this method only takes into account completed connections.

Fig. 5.19: Network Topologies for SPN Network Model.

To illustrate the results, two different network topologies are considered and analyzed in this thesis. The first topology consists of four nodes and six links whereas the second topology consists of five nodes with ten links. It is believed that these topologies are sufficiently complex to be representative of backbone networks. Each connection in the backbone network can carry multiple services that are grouped together based on priority levels. The resources on one connection can be shared by different users. In the event of unforeseen link failure or node failure, preemption can provide an extra layer of protection by giving resources to the affected high priority connections. The two topologies used for analysis are shown in Fig. 5.19. The

primary purpose of this analysis is to capture the impact of preemption on network performance.

Connection requests originated from different node-pairs are shown in Table 5.4 and Table 5.5 respectively. For each node-pair, the connection will first use the shortest path before using the longer one. Similarly, if preemption is required, the connection will first try to preempt on the shortest path. Four priority levels are considered which range from priority 0 to 3. The capacity for each of the links is 8 units and a connection can request three different bandwidth levels, i.e. 1, 2, and 4 units respectively. The firing rate of *Server* is 0.25 connections/s. The firing rate of *Arr* is varied from 0.1 to 0.42 connections/s. These network parameters have been chosen to be representative of a backbone network while allowing the application of the SPN technique to study the behavior of the preemption strategy. The model is generated and solved by using SPNP software [89].

For the 4-node topology, six copies of link model are placed to represent the network links and 25 copies of depository to store connections information. The number of immediate transitions used for node-pair selection is six, i.e., from *np1* to *np6*. Likewise, ten copies of link model and 35 copies of depository are used for the 5-node topology. Ten immediate transitions are used for node-pair selection, from *np1* to *np10*.

The firing probability are assigned equally to the four priority levels such that a connection is equally likely to be assigned one of the four priority levels. The results for ‘priority’ are based on the simple preemption approach presented in Section 5.4.2, whereas ‘PREM’ represents the approach in [80]. ‘SEP’ represents the scheme proposed in Chapter 4.

Table 5.4: Node Pairs and Links Used for 4-Node Topology.

Node Pair	Links Used
(1,2)	(a), (d,f)
(1,3)	(e), (a,b), (d,c)
(1,4)	(d), (a,f)
(2,3)	(b), (f,c)
(2,4)	(f), (b,c), (a,d)
(3,4)	(c), (b,f)

Table 5.5: Node Pairs and Links Used for 5-Node Topology.

Node Pair	Links Used
(1,2)	(b), (g,h)
(1,3)	(i), (b,c)
(1,4)	(h), (a,e)
(1,5)	(a), (i,j)
(2,3)	(c), (f,j)
(2,4)	(g), (c,d)
(2,5)	(f), (a,b)
(3,4)	(d), (i,h)
(3,5)	(j), (d,e)
(4,5)	(e), (f,g)

Table 5.6: Sizes of the Markov Chains for the Network Model.

Model	No. States	No. Transitions	No. Tangible Markings	No. Non Zero Entries	Time Taken
4-node No Preemption	180	175	6,241	1,102,819	~1 hr
4-node Prio	211	194	6,357	1,216,738	~1 hr
4-node PREM	223	205	6,554	1,378,215	~1 hr
4-node SEP	235	210	6,736	1,426,346	~ 1 hr
5-node No Preemption	254	236	11,548	2,261,755	~2 hrs
5-node Prio	296	282	12,367	2,523,294	~2 hrs
5-node PREM	313	295	12,931	2,825,512	~2 hrs
5-node SEP	337	302	13,248	3,476,322	~ 3hrs

Fig. 5.20: Numerical and Simulation Results on Network Throughput for 5 Node Network.

Fig. 5.20 shows the comparison between the results obtained numerically using SPNP and simulation written in C++. Both sets of results give similar performances which indicate the feasibility of the SPN model developed.

Fig. 5.21 and Fig. 5.22 show the network throughput for the 4-node and 5-node network, respectively. In the figures, network throughput increases linearly with the offered load but starts to saturate at higher offered. Networks with preemption system show lower throughput chiefly because the throughput of preempted connections are not accrued. In addition, more than one connection could be preempted by a new arrival. 'SEP' produces the highest throughput as it is able to re-route preempted connections. However, the improvement is not highly significant as the number of re-routable paths is limited. 'PREM' performs better than 'priority' due to its effectiveness in reducing excessive bandwidth preempted.

Fig. 5.21: Network Throughput for 4 Node Network.

Fig. 5.22: Network Throughput for 5 Node Network.

Fig. 5.23: Preemption Probability.

Preemption probability is illustrated in Fig. 5.23. It is observed that initially there is little difference in performance between the different preemption schemes. However, the preemption probability for 4-node topology is higher than 5-node topology at higher arrival rates. This is due to the fact that the 4-node topology has a smaller number of links and the resources may get fully utilized more rapidly leading to higher preemption. Furthermore, for both topologies, ‘PREM’ shows the lowest preemption probability due to its ability to minimize bandwidth preempted. ‘SEP’ produces the highest preemption probability because re-routing mechanism allows the networks to carry a higher number of connections which in turn increases the probability that a new connection will trigger preemption. Preemption probability stabilizes at a higher offered load because the new incoming connections will find it harder to trigger preemption as the lower priority connections are constantly

preempted. The higher priority connections dominate the network so that new connections cannot preempt and hence the preemption probability saturates. With high preemption probability, services for low priority connections will be greatly affected. This implies that preemption system is more likely to be suitable for network loads ranging from medium to high levels.

In Fig. 5.24 and Fig. 5.25, analysis is conducted for the ‘priority’ case in the 5-node topology and biased the connection priority such that a connection is assigned to priority 0, 1, 2, and 3 with probability 0.4, 0.3, 0.2 and 0.1, respectively. In Fig. 5.24, the network throughput with biased probability is lower due to the fact that more existing connections are preempted as many connections are assigned with higher probability. However, at higher arrival rates, the network throughput with biased probability becomes higher because high priority connections quickly dominate the network and subsequently reduce the preemption events. In Fig. 5.25, it is apparent that preemption probability is higher at low arrival rates for the biased priority case. On the other hand, at high arrival rates, the reverse is true.

Fig. 5.24: Network Throughput for 'Priority' Case in the 5-Node Topology.

Fig. 5.25: Preemption Probability for 'Priority' Case in the 5-Node Topology.

5.5 Summary

This Chapter presents a new generic SPN based approach for modeling networks with preemption schemes and studies its numerical performance. It can serve as a common platform to evaluate the performance of different preemption strategies and would be useful in studying connection oriented networks, e.g. MPLS. In the future work, various decomposition techniques can be incorporated so that the approach may be extended to model networks of even higher complexity. Some of the existing decomposition techniques are presented in [97], in which a SPN model is scrutinized to identify the decomposable substructures. A decomposed model can be solved more efficiently and may overcome the problem of computational limitation.

To extend the investigation on preemption issues, sensitivity analysis and bottleneck detection [98-99] can be performed on the SPN model. For example in a network preemption scenario, analysis can be conducted to investigate the effect of a bottleneck link on the network throughput. The SPNP package is able to compute the measures of interest with respect to input parameters. It is also possible to conduct transient analysis [100] on the SPN model to quantify the delay associated with preemption. Furthermore, more elaborative form of routing [101] that depends on network congestion can be incorporated to enhance the capability of the model.

6. Conclusion and Recommendations

In this chapter, the author summarizes the work reported in this thesis. In addition, several possible directions are discussed for further research.

6.1 Conclusion

The main focus of this work is the preemption issues in the connection-oriented networks. Preemption issues arise from network resources competition which may be due to oversubscription, node failure or link failure. In order to provide available and reliable services to high priority connections under limited resources environment, preemption mechanism tears down a number of existing low priority connections to free up sufficient resources needed. However, by triggering preemption, end users of preempted connections will experience service disruption. This could affect users' perception on the network services provided.

In Chapter 3, the author proposes the preemption strategy with re-routing to minimize service disruption. By re-routing the to-be-preempted connections, ongoing connections will be unaffected. To realize the full potential of re-routing, we design the strategy such that re-routable connections are preempted before the non re-routable connections. As such, the network will attempt to re-route as many

connections as possible in each preemption event. Simulation results show that service disruption is greatly reduced as compared to existing strategies that do not incorporate re-routing. The decentralized strategy proposed simplifies the implementation but with comparable performance.

In Chapter 4, the author shows that active network management through preemption is able to improve network performance. In order to reduce the resources consumed by high priority connections, we use preemption more actively to route the connections consistently on the shortest paths. This is in contrast to the traditional approach whereby a connection will search for all possible paths before triggering preemption. Given that high priority connections are not subject to preemption, by routing high priority connections on the shortest paths, this strategy improves the admission rate of future connections as well as the network throughput. However, with higher preemption rates, more re-routing events will happen. A re-routing control technique is designed that only allows re-routing if the alternative path consumes less than or equal to the resources consumed by the original path. Simulation results confirm that network throughput is greatly improved using this strategy.

In Chapter 5, the Stochastic Petri Net (SPN) models for preemption are proposed. The single link preemption model as well as the network preemption model can be used as a common platform to compare the performance of different preemption strategies. They can also be used to validate various preemption strategy developed. These models are designed such that it can be easily expanded to analyze network of higher complexity.

6.2 Recommendations for Future Research

In all of the work, it is assumed that no further preemption is triggered when the network is re-routing the existing connections. By allowing the cascading of preemption events, network stability may be affected. However, the main advantage is that the network is able to better allocate more resources to high priority connections. Therefore, the tradeoff between more preemption events and network stability will be an important issue in future research. In addition, we would like to implement the preemption strategies in real MPLS test-bed. As the strategies formulated encompass both centralized and decentralized approaches, this actual implementation will provide insights to the performance of both approaches. It can also be used to verify the results observed in simulations.

Although the SPN models developed are easily expandable, it may grow to a size that could strain the computational power of the current state-of-the-art machines. Therefore, it is vital to decompose the models such that the networks of higher complexity are solvable in acceptable duration. Furthermore, this decomposition may reveal some other important features of preemption. The future models to be developed may include more flexible routing (as opposed to the current fixed routing).

Author's Publications

- [1] C.H. Lau, B.H. Soong and S.K. Bose, "Preemption algorithm with re-routing to minimize service disruption," in *Ninth Int'l Conf. Communications Systems*, Singapore, 2004, pp. 521-525.
- [2] C.H. Lau, B.H. Soong and S.K. Bose, "SERA: Service ratio based preemption for minimizing throughput loss in networks," in *IEEE Region 10 Conf. TENCN*, Melbourne, 2005, pp. 1-6.
- [3] C.H. Lau, B.H. Soong and S.K. Bose, "A new preemption strategy with re-routing control for MPLS networks," in *Fifth IEEE Intl. Conf. Information, Communications and Signal Processing*, Bangkok, 2005, pp. 1515-1519.
- [4] C.H. Lau, B.H. Soong and S.K. Bose, "Preemption with re-routing to minimize service disruption in connection-oriented networks," *IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans*, accepted for publication.
- [5] C.H. Lau, B.H. Soong and S.K. Bose, "Path selection with preemption and re-routing control for MPLS networks," *Computer Communications*, vol. 29, no. 18, pp. 3718-3732, Nov. 2006.
- [6] C.H. Lau, B.H. Soong and S.K. Bose, "Modeling a network preemption system for connection-oriented networks using stochastic Petri nets," submitted for publication.

Bibliography

- [1] X. Xiao and L. M. Ni, "Internet QoS: a big picture," *IEEE Network Magazine*, vol. 13, no. 2, pp. 8-18, Mar-Apr, 1999.
- [2] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource reservation protocol (RSVP)," IETF RFC 2205, Sept. 1997.
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, Dec. 1998.
- [4] W. C. Hardy, *QoS Measurement and Evaluation of Telecommunications Quality of Service*, New York: Wiley, 2001.
- [5] H. J. Chao and X. Guo, *Quality of Service Control in High-Speed Networks*, New York: Wiley, 2002.
- [6] Y. Bernet, P. Ford, R. Yavatkar, F. Baker *et al.*, "A Framework for integrated services operation over Diffserv networks," IETF RFC 2998, Nov. 2000.
- [7] S. Shenker, C. Partridge and R. Guerin, "Specification of guaranteed quality of service," IETF RFC 2212, Sept. 1997.
- [8] J. Wroclawski, "Specification of the controlled-load network element service," IETF RFC 2211, Sept. 1997.
- [9] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the internet," IETF RFC 2638, Jul. 1999.
- [10] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 header," IETF RFC 2474, Dec. 1998.
- [11] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured forwarding PHB group," IETF RFC 2597, Jun. 1999.

-
- [12] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," IETF RFC 2598, Jun. 1999.
 - [13] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, Jan. 2001.
 - [14] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: extensions to RSVP for LSP tunnels," IETF RFC 3209, Dec. 2001.
 - [15] B. Jamoussi, L. Anderson, R. Callon, R. Dantu, L. Wu, P. Doolan, *et al.*, "Constraint-based LSP setup using LDP," IETF RFC 3212, Jan. 2002.
 - [16] J. Lawrence, "Designing Multiprotocol Label Switching Networks," *IEEE Communications Magazine*, vol. 39, no. 7, pp. 134-142, Jul. 2001.
 - [17] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McAnus, "Requirements for traffic engineering over MPLS," IETF RFC 2702, Sept. 1999.
 - [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] B. Carpenter and K. Nichols, "Differentiated services in the internet," in *Proc. IEEE*, 2002, vol. 90, no. 9, pp. 1479-1494.
 - [20] K. Phanse and L. DaSilva, "Addressing the requirements of QoS Management for wireless ad hoc networks," *Computer Communications*, vol. 26, no. 12, pp. 1263-1273, Jul. 2003.
 - [21] S. Choi and K. G. Shin, "A cellular wireless local area network with QoS guarantees for heterogeneous traffic," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 89 – 100, Jun. 1998.
 - [22] G. Anastasi and L. Lenzini, "QoS provided by the IEEE 802.11 wireless LAN to advanced data applications: a simulation analysis," *Wireless Networks*, vol. 6, no. 2, pp. 99–108, 2000.
 - [23] H. S. Hassanein, H. Chen, and H. T. Mouftah, "A service model for guaranteeing packet loss bounds in differentiated services architectures," *Computer Communications*, vol. 26, no. 10, pp. 1083-1094, Jun. 2003.

-
- [24] S. C. Borst, R. Nunez-Queija, and M. J. G. van Uiter, "User-level performance of elastic traffic in a differentiated-services environment," *Performance Evaluation*, vol. 49, no. 1-4, pp 507-519, Sept. 2002.
 - [25] A. Bak, W. Burakowski, F. Ricciato, S. Salsano, and H. Tarasiuk, "A framework for providing differentiated QoS guarantees in IP-based network," *Computer Communications*, vol. 26, no. 4, pp 327-337, Mar. 2003.
 - [26] I. Mahadevan, and K. M. Sivalingam, "Architecture and experimental framework for supporting QoS in wireless networks using differentiated services," *Mobile Networks and Applications*, vol. 6, no. 4, pp 385-395, Aug. 2001.
 - [27] Y. Bernet, "The complementary roles of RSVP and differentiated services in the full-service QoS network," *IEEE Communications Magazine*, vol. 38, no. 2, pp. 154-162, Feb. 2000.
 - [28] F. M. Serenato and J. Rochol, "MPLS backbone and QoS enabled networks interoperation," *Proc. 2003 IFIP/ACM Latin America Conf. Towards Latin America Agenda for Network Research*, La Paz, Bolivia, pp. 44-57.
 - [29] N. Feamster, J. Borkenhagen and J. Rexford, "Guidelines for interdomain traffic engineering," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 5, pp. 19-30, Oct. 2003.
 - [30] M. Roughan, M. Thorup and Y. Zhang, "Traffic engineering with estimated traffic matrices," *Proc. 3rd ACM SIGCOMM Conf. Internet Measurement*, Miami, 2003, pp. 248-258.
 - [31] S. Kandula, D. Katabi, B. Davie and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," *Proc. 2005 Conf. Applications, Tech., Architectures, and Protocol Computer Comm.*, Philadelphia, pp. 253-264.
 - [32] R. Teixeira, T. G. Griffin, M. Resende, and J. Rexford, "Tie breaking: tunable interdomain egress selection," *Proc. 2005 ACM Conf. Emerging Network Experiment and Technology*, Toulouse, pp. 93-104.
 - [33] C. Vogt, "Admission control and resource reservation on the internet," *AGM SIGSOFT Software Engineering Notes*, vol. 27, no. 3, pp. 80-87, May 2002.

-
- [34] L. Breslau, E. Knightly, S. Shenker, I. Stoica and H. Zhang, "Endpoint admission control: architectural issues and performance," *Proc. Conf. Applications, Tech., Architectures and Protocols Computer Comm.*, Stockholm, 2000, pp. 57-69.
- [35] C. Cetinkaya and E. W. Knightly, "Egress admission control," *Proc. Nineteenth Annual Joint Conf. IEEE Computer and Comm. Societies, INFOCOM 2000*, Tel Aviv, vol. 3, pp. 1471-1480.
- [36] R. Hill, and H.T. Kung, "A Diff-Serv enhanced admission control scheme," *IEEE Global Telecommunications Conf., Globecom*, San Antonio, 2001, vol. 4, pp. 2549-2555.
- [37] C. Lindemann, M. Lohmann and A. Thummler, "Adaptive call admission control for QoS/revenue optimization in CDMA cellular networks," *Wireless Networks*, vol. 10, no. 4, pp. 457-472, Jul. 2004.
- [38] S. Wu, K. Y. Michael Wong, and B. Li, "A dynamic call admission policy with precision QoS guarantee using stochastic control for mobile wireless networks," *IEEE/ACM Trans. on Networking*, vol. 10, no. 2, pp. 257-271, Apr. 2002.
- [39] S. Liu, J. Gu, and B. Gong, "Improved load management based on resources reservation in distributed embedded systems," *Intl. Conf. Machine Learning and Cybernetics*, 2003, vol. 4, pp. 2015-2019.
- [40] S. Ardon, C. Diot, B. Landfeldt, and A. Seneviratne, "Resources reservation in a reactive QoS scheme," *Global Telecommunications Conf., Globecom*, Rio de Janeiro, 1999, vol. 1b, pp. 817-824.
- [41] G. Dermler, W. Fiederer, I. Barth and K. Rothermel, "A negotiation and resource reservation protocol (NRP) for configurable multimedia applications," *Proc. 1996 Intl. Conf. Multimedia Computing and Systems, ICMCS*, 1996, pp. 0113.
- [42] S. Tong, Y. Yu, C. Huang, and M. Lin, "Efficient resource reservation based on communication paradigms for multicast multimedia applications," *IEEE Trans. on Broadcasting*, vol. 50, no. 3, pp. 260-278, Sept. 2004.

-
- [43] H. A. Mantar, J. Hwang, I. T. Okumus and S. J. Chapin, "A scalable model for interbandwidth broker resource reservation and provisioning," *IEEE J. Selected Areas in Communications*, vol. 22, no. 10, pp. 2019-2034, Dec. 2004.
- [44] M Mahajan, A. Ramanathan and M. Parashar, "Active resource management for the differentiated services environment," *Int'l J. Network Management*, vol. 14, no. 3, pp. 149-165, May 2004.
- [45] A. G. Greenberg, R. Srikant, and W. Whitt, "Resource sharing for book-ahead and instantaneous-request calls," *IEEE/ACM Trans. Networking*, vol. 7, no. 1, pp. 10-22, Feb. 1999.
- [46] I. Ahman, J. Kamruzzaman, and S. Aswathanarayanan, "Dynamic look-ahead time in book-ahead reservation in Qos-enabled networks," *Proc. 12th IEEE Int'l Conf. Networks*, 2004, vol. 2, pp. 566-571.
- [47] I. Ahmad, J. Kamruzzaman, and S. Aswathanarayanan, "Dynamic look-ahead time with conditional call admission based on predicted lifetime," *Ninth Intl. Conf. Communications Systems*, Singapore, 2004, pp. 537-541.
- [48] Y. Lin, C. Chang, and Y. Hsu, "Bandwidth brokers of instantaneous and book-ahead requests for differentiated services networks," *ICICE Trans. Communications*, vol. E85-B, no. 1, pp. 278-283, 2002.
- [49] R. Guerin *et al*, "Network with advance reservations: the routing perspective," *Proc. Nineteenth Annual Joint Conf. IEEE Computer and Communications Societies*, Tel Aviv, 2000, vol. 1, pp. 118-127.
- [50] N.J. Keon, and G. Anandalingam, "Optimal pricing for multiple services in telecommunications networks offering quality-of-service guarantees," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 66-80, Feb. 2003.
- [51] C. Frei, B. Faltings, and M. Hamdi, "Resource allocation in communication networks using abstraction and constraint satisfaction," *IEEE J. Selected Areas Communications*, vol. 23, no. 2, pp. 304-320, Feb. 2005.
- [52] A. R. Bashandy, E. K. P. Chong, and A. Ghafoor, "Generalized quality-of-service routing with resource allocation," *IEEE J. Selected Areas Communications*, vol. 23, no. 2, pp. 450-463, Feb. 2005.

-
- [53] X. Yuan, "Heuristics algorithms for multiconstrained quality-of-service routing," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 244-256, Apr. 2002.
- [54] Z. Wang, and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE J. Selected Areas Communications*, vol. 14, no. 7, pp. 1228-1234, Sept. 1996
- [55] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum interence routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE J. Selected Areas Communications*, vol. 18, no. 12, pp. 2566-2579, Dec. 2000.
- [56] A. B. Bagula, M. Botha, and A. E. Krzesinski, "Online traffic engineering: the least interference optimization algorithm," *IEEE Int'l Conf. Communications*, 2004, vol. 2, pp. 1232-1236.
- [57] A. Capone, and F. Martignon, "Analysis of dynamic Qos routing algorithms for MPLS networks," *IEEE Int'l Conf. Communications*, 2004, vol. 2, pp. 1192-1196.
- [58] K. Hendling, T. Losert, W. Huber, and M. Jادل, "Interference minimizing bandwidth guaranteed on-line routing algorithm for traffic engineering," *Proc. 12th IEEE Int'l Conf. Networks, ICON*, 2004, vol. 2, pp. 497-503.
- [59] B. Szviatovszki, A. Szentesi, and A. Juttner, "Minimizing re-routing in MPLS networks with preemption-aware constraint-based routing," *Computer Communications*, vol. 25, no. 11, pp 1076-1084, Jul. 2002.
- [60] F. Blanchy, L. Mélon, and G. Leduc, "Routing in MPLS network featuring preemption mechanisms," *10th Int'l Conf. Telecommunications*, 2003, vol. 1, pp. 253-260.
- [61] I. Ahmad, J. Kamruzzaman, and S. Aswathanarayanan, "Preemption-aware routing for QoS-enabled networks," *IEEE Global Telecommunications Conf., Globecom*, St. Louis, 2005, vol. 2, pp. 6.

-
- [62] J. L. Marzo, E. Calle, C. Scoglio and T. Anjali, "QoS online routing and MPLS multilevel protection: a survey," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 126-132, Oct. 2003.
 - [63] J. L. Marzo, E. Calle, C. Scoglio and T. Anjali, "Adding QoS protection in order to enhance MPLS QoS routing", *IEEE Int'l Conf. Communications, ICC*, 2003, vol. 3, pp. 1973-1977.
 - [64] Y. Bejerno, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, "Algorithms for computing QoS paths with restoration," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 648-661, Jun. 2005.
 - [65] K. Kar, M. Kodialam, and T. V. Lakshman, "Routing restorable bandwidth guaranteed connections using maximum 2-route flows," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 772-781, Oct. 2003.
 - [66] J. Li and K. L. Yeung, "A novel two-step approach to restorable dynamic QoS routing," *J. Lightwave Technology*, vol. 23, no. 11, pp. 3663-3670, Nov. 2005.
 - [67] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, "Routing bandwidth guaranteed paths with local restoration in label switched networks," *IEEE J. Selected Areas Communications*, vol 23, no 2, pp. 437-449, Feb. 2005.
 - [68] S. Raza, F. Aslam, and Z. A. Uzmi, "Online routing of bandwidth guaranteed paths with local restoration using optimized aggregate usage information," *IEEE Int'l Conf. Communications, ICC*, 2005, vol. 1, pp. 201-207.
 - [69] L. Melon, F. Blanchy, and G. Leduc, "Decentralized local backup LSP calculation with efficient bandwidth sharing", *10th Int'l Conf. Telecommunications, ICT*, 2003, vol. 2, pp.929-937.
 - [70] J. A. Garay, and I. S. Gopal, "Call preemption in communication networks", *Eleventh Annual Joint Conf. IEEE Computer and Communications Societies, INFOCOM*, Florence, 1992, vol. 3, pp. 1043-1050.
 - [71] M. Garey, and D. Johnson, *Computer and Intractability, a Guide to the Theory of NP-Completeness*, New York: W.H. Freeman & Company, 1979.

-
- [72] M. Peyravian, and A. D. Kshemkalyani, "Decentralized network connection preemption algorithms," *Computer Networks and ISDN Systems*, vol. 30, no. 11, pp. 1029 – 1043, Jun. 1998.
- [73] M. Peyravian, "Providing different levels of network availability in high-speed networks," *IEEE Global Telecommunications Conf., Globecom*, San Francisco, 1994, vol. 2, pp. 941-945.
- [74] S. Poretsky, "Connection precedence and preemption in military asynchronous transfer mode (ATM) networks," *Proc. IEEE Military Communications Conf., MILCOM*, 1998, vol. 1, pp. 86-90.
- [75] S. Poretsky and T. Gannon, "An algorithm for connection precedence and preemption in ATM networks," *IEEE Int'l Conf. Communications, ICC*, 1998, vol.1, pp. 299-303.
- [76] A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, and B. Schieber, "Bandwidth allocation with preemption," *SIAM J. Computing*, vol. 28, no. 5, pp. 1806–1828, Apr/May 1999.
- [77] Y. Z. Cho, and C. K. Un, "Analysis of the M/G/1 queue under a combined preemptive/nonpreemptive priority discipline," *IEEE Trans. Communications*, vol. 41, no 1, pp 132-141, Jan. 1993.
- [78] J. Wang, Q. Zeng, and D. P. Agrawal, "Performance analysis of a preemptive and priority reservation handoff scheme for integrated service-based wireless mobile networks," *IEEE Trans. Mobile Computing*, vol. 2, no. 1, pp. 65-75, Jan. 2003.
- [79] W. Li, H. Chen, and D. P. Agrawal, "Performance analysis of handoff schemes with preemptive and nonpreemptive channel borrowing in integrated wireless cellular networks," *IEEE Trans. Wireless Communications*, vol. 4, no. 3, pp. 1222-1233, May 2005.
- [80] J. C. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl, "A new preemption policy for DiffServ-aware traffic engineering to minimize rerouting," *Twenty-First Annual Joint Conf. IEEE Computer and Communications Societies, Infocom*, 2002, vol. 2, pp. 695-704.

-
- [81] J. C. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl, "New preemption policies for DiffServ-aware traffic engineering to minimize rerouting in MPLS networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 4, pp. 733-745, Aug. 2004.
 - [82] R. C. Vieira, and P. R. Guardieiro, "A proposal and evaluation of a LSP preemption policy implemented with fuzzy logic and genetic algorithms in a Diffserv/MPLS test-bed," *Proc. Int'l Conf. Communications, Circuits, and Systems*, 2005, vol. 1, pp. 109-114.
 - [83] V. Stanisic, and M. Devetsikiotis, "A dynamic study of providing quality of service using preemption policies with random selection," *IEEE Int'l Conf. Communications, ICC*, 2003, vol. 3, pp. 1543-1546.
 - [84] S. Jeon, R. T. Abler, and A. E. Goulart, "The optimal connection preemption algorithm in a multi-class network," *IEEE Int'l Conf. Communications, ICC*, 2002, vol. 4, pp. 2294-2298.
 - [85] T. Shan, H. Dam, O. Yang, "Bandwidth allocation and preemption for supporting differentiated-service-aware traffic engineering in multi-service networks," *IEEE Int'l Conf. Communications*, 2002, vol. 2, pp. 1305-1309.
 - [86] Li Lei, and S. Sampalli, "Backward connection preemption in multi-class QoS-aware networks," *Twelfth IEEE Int'l Workshop Quality of Service*, 2004, pp. 153-160.
 - [87] P. Dharwadkar, H. J. Siegel, and E. K. P. Chong, "A heuristic for dynamic bandwidth allocation with preemption and degradation for prioritized requests," *21st Int'l Conf. Distributed Computing Systems*, 2001, pp. 547-556.
 - [88] M. R. Meyer, D. Maddux, J.-P. Vasseur, C. Villamizar, and A. Birjandi, "MPLS traffic engineering soft preemption", IETF RFC Draft, Oct 2003.
 - [89] G. Ciardo, K.S. Trivedi and J. Muppala, "SPNP: stochastic Petri net package," *Proc. 3rd Int'l Workshop Petri Nets and Performance Models, PNPM89*, Kyoto, 1989, pp. 142-151.
 - [90] R. K. Ahuja, T.L. Magnant and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Englewood Cliffs, N.J.:Prentice Hall, 1993.

-
- [91] B. Teitelbaum, "QBone architecture," Internet2 QoS Working Group Draft, Aug. 1999.
- [92] D. A. Dunn, W. D. Grover, and M. H. MacGregor, "Comparison of k -shortest paths and maximum flow routing for network facility restoration," *IEEE J. Selected Areas Communication*, vol. 12, no. 1, pp. 88-99, Jan. 1994.
- [93] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," *Proc. Int'l Conf. Network Protocols*, Atlanta, 1997, pp. 191-202.
- [94] J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, Englewood Cliffs, N.J.:Prentice Hall, 1981.
- [95] E. Smeitink, and R. Dekker, "A simple approximation to the renewal function," *IEEE Trans. Reliability*, vol. 39, no. 1, pp. 71-75, Apr. 1990.
- [96] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, New York:Wiley, 2001.
- [97] G. Ciardo and K.S. Trivedi, "A decomposition approach for stochastic reward net models," *Performance Evaluation*, vol. 18, no. 1, pp. 37-59, Jul. 1993.
- [98] J.T. Blake, A.L. Reibman, and K.S. Trivedi, "Sensitivity analysis of reliability and performability measures for multiprocessor systems," *Proc. 1998 ACM SIGMETRICS Conf. Measurements and Modeling of Computer Systems*, Santa Fe, New Mexico, 1998, pp. 177- 186.
- [99] J.K. Muppala and K.S. Trivedi, "GSPN models: sensitivity analysis and applications," *Proc. 28th ACM Southeast Region Conf.*, Greenville, SC, 1990, pp. 24-33.
- [100] H. Sun, J. J. Han, and H. Leveldel, "Transient analysis for prioritized failure recovery in communication networks," *21st IEEE Int'l Conf. Performance, Computing and Communications*, Phoenix, 2002, pp. 213-219.
- [101] M. Balakrishnan, and K.S. Trivedi, "Stochastic petri nets for the reliability analysis of communication network applications with alternate routing," *Reliability Engineering and System Safety*, vol. 52, no. 3, pp 243 – 259, 1996.

Appendix I

A random variable X has a Coxian-2 distribution if

$$X = \begin{cases} X_1 & \text{with probability } p \\ X_1 + X_2 & \text{with probability } 1 - p \end{cases}$$

where $0 \leq p \leq 1$, X_1 and X_2 are exponentially distributed with mean $1/\lambda_1$ and $1/\lambda_2$.

The pdf for Coxian-2 distribution is,

$$f(t) = \begin{cases} p\lambda \exp\{-\lambda t\} + (1-p)\lambda^2 t \exp\{-\lambda t\}, & \lambda_1 = \lambda_2 = \lambda \\ \frac{p\lambda_1 - \lambda_2}{\lambda_1 - \lambda_2} \lambda_1 \exp\{-\lambda_1 t\} + \left(1 - \frac{p\lambda_1 - \lambda_2}{\lambda_1 - \lambda_2}\right) \lambda_2 \exp\{-\lambda_2 t\}, & \lambda_1 \neq \lambda_2 \end{cases}$$

The parameters can be selected such that:

$$\lambda_1 = \frac{2}{E\{X\}} \left(1 + \sqrt{\frac{c_T^2 - 1/2}{c_T^2 + 1}}\right), \quad \lambda_2 = \frac{4}{E\{X\}} - \lambda_1$$

$$p = (1 - \lambda_2 E\{X\}) + \lambda_2 / \lambda_1$$

The pdf for $E_{n-1,n}$ distribution is given by:

$$f(t) = p\lambda^{n-1} \frac{t^{n-2}}{(n-2)!} \exp\{-\lambda t\} + (1-p)\lambda^n \frac{t^{n-1}}{(n-1)!} \exp\{-\lambda t\}, \quad t \geq 0$$

The parameters can be obtained by,

$$\frac{1}{n} \leq c_T^2 \leq \frac{1}{n-1}, \quad \lambda = \frac{n-p}{E\{X\}}, \quad p = \frac{1}{c_T^2} \left\{ nc_T^2 - [n(1+c_T^2) - n^2 c_T^2]^{1/2} \right\}$$