ABSTRACT

STANISIC, VLADICA Application Based Resource Allocation Policies in MultiService Networks. (Under the direction of Associate Professor Mihail Devetsikiotis).

Efficient and reliable bandwidth allocation is one of the most important open issues in the management of networks that aim to offer a guaranteed Quality of Service. The bandwidth allocation problem becomes more difficult in multi-service networks, where a large variety of different applications, each one with different requirements in terms of bandwidth, duration or delay, information loss use the network infrastructure simultaneously. Most of the previous work has analyzed bandwidth allocation policies under the context of resolving conflicts due to dynamics of user requests without taking network availability, user mobility, or the delivery (i.e., physical environment) conditions into account. Since static bandwidth allocation policies lack adaptive mechanisms to combat these dynamics in the network and improve bandwidth utilization, we believe that a more flexible service model which allows variable QoS is needed. Adaptive resource management coupled with dynamic load balancing aims at decreasing the possibility of congestion and maintaining high resource utilization, under transient traffic variations and node/link failure. We have formulated preemption algorithms and criteria for optimization by preemption algorithms, studied existing algorithms and investigated sub-optimal preemption algorithms with random selection of connections to be rerouted. We have also performed numerical and simulation comparisons of re-routing algorithms by analyzing their performance on a single link, dynamic setting and in a full network environment with a heterogeneous traffic mix. In order to account for the users application type, QoS requirements and quantify users' value we have introduced a utility-based QoS model. We have investigated network utilization, QoS observed by the customers, and revenue generation perspectives for different utility-quantified bandwidth allocation schemes. We have presented approximate analytical tools to obtain blocking probabilities in a multi rate multi class system, where users of the same class can have different resource requirements. We have evaluated the blocking probabilities for a single link case and validated our approach through the simulation of such a system. Also we have expanded our single link model to calculate blocking probabilities for a multihop path, when the offered traffic of each source destination pair along the path is known.

Application Based Resource Allocation Policies in MultiService Networks

by

Vladica Stanisic

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Department of Electrical Engineering

Raleigh

2005

Approved By:

Dr. Mladen A. Vouk	Dr. J Keith Townsend
Dr. Mihail Devetsikiotis Chair of Advisory Committee	Dr. Arne A. Nilsson

To my family.

Biography

I was born in Belgrade, Serbia and Montenegro, in 1975. I have attended the University of Belgrade, and graduated in Electrical Engineering in 1999. Since 2000 I have been attending North Carolina State University.

Acknowledgements

I would like to thank my advisor Dr. Michael Devetsikiotis for his guidance during each stage of this research. I would also like to thank Dr. Arne Nilsson, Dr. Mladen Vouk and Dr. Keith Townsend for serving on my committee and their expert advice and valuable suggestions. I am also grateful to Dr. Yannis Vinniotis and Dr. Khaled Harfoush for their profound insights into the world of computer networks in the course of my studies.

And finally, I would like to express my wholehearted gratitude to my family and friends for their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Efficient and reliable bandwidth allocation is one of the most important open issues in the management of networks that aim to offer a guaranteed Quality of Service (QoS). The bandwidth allocation problem becomes more difficult in multi-service networks, where a large variety of different applications, each one with different requirements in terms of bandwidth, duration or delay and information loss, use the network infrastructure simultaneously. The task of efficient bandwidth allocation is further complicated when one introduces QoS in an environment of mobile hosts, wireless networks, and different access technologies with dynamically changing topologies and resources. Yet, the need for QoS mechanisms in time-varying and bandwidth limited networks is greater due to scarce resources, unpredictable available bandwidth and variable error rates.

1.1 Motivation

An introduction of QoS in the network provides benefits of improved reliability and predictability of the service but the consequence is increase in complexity, management overhead and effort for additional configuration. Traffic engineering functions are mainly concerned with the management of network resources with the purpose of accommodating offered traffic while optimizing network resource utilization and traffic performance. They aim to maximize the network resources utilization (improve flexibility of the network), while minimizing the number of connections that would be denied access due to insufficient resource availability. *Load balancing* is another important issue. It is desirable to avoid that portions of the network become over-utilized and congested, while alternative feasible paths remain under-utilized. A good bandwidth allocation strategy is fundamental for such networks.

In networks that aim to offer a guaranteed QoS policy, if a user (e.g., an application or connection aggregate) requests a guaranteed bandwidth for a communication, the user has to reserve, in advance, the amount of bandwidth required. The advantages of a guaranteed bandwidth policy are many:

1. Bounded delay: the delay experienced by user is bounded for real-time applications.

2. Differentiated service: users can expect different levels of quality of service (QoS) based on the pricing of network service and type of service requested.

3. Simple pricing: a user can be charged based on the amount of allocated bandwidth.

4. Fairness: one user cannot occupy all the available bandwidth [14].

The nature of the bandwidth allocation problem is such that a decision made previously to accept a connection may have been wrong because it caused a future "more valuable" connection request to be rejected. For example, an accepted connection may have used up capacity from the network , thereby causing the connection of much larger bandwidth requirement (consequently larger revenue to the network) to be rejected.

Most of the previous work [36, 30, 11, 8] has analyzed bandwidth allocation policies under the context of resolving conflicts due to dynamics of user requests without taking network availability, user mobility, or the delivery (i.e., physical environment) conditions into account. Since static bandwidth allocation policies lack adaptive mechanisms to combat these dynamics in the network and improve bandwidth utilization, we believe that a more flexible service model which allows *variable QoS* is needed. The major drawback of the static guaranteed bandwidth policies is inefficiency for the applications that cannot accurately estimate their traffic parameters when the flow is first established or for flows with rate variations over multiple time scales. Resources are wasted when certain users are not offering much traffic to the network, because the capacity reserved for them goes unused.

Due to statistical fluctuations, user mobility or occasional unavailability of resources (due to faults or attacks), some connections that could otherwise have been accepted if the traffic load were better balanced (with consequently larger revenue to the network), are instead rejected. Adaptive resource management coupled with dynamic load balancing aims at decreasing the possibility of congestion and maintaining high resource utilization, under transient traffic variations and node availability.

There are several challenges and technical difficulties that have prevented so far the solution to the problems mentioned above. Such challenges include:

- Absence of a existing bandwidth allocation scheme which incorporates the utility of the connections and accounts for the users QoS requirements.
- Given that dynamic bandwidth allocation is a real-time problem and its online nature (a decision whether to accept a new request cannot be delayed), a bandwidth allocation calculation must satisfy strict constraints in terms of both average user-perceived performance and complexity.
- Difficulty in combining utilization and fairness in order to balance achieving high bandwidth utilization and fairness among users [15].

1.2 Thesis Contributions

We have formulated preemption algorithms and criteria for optimization by preemption algorithms, studied existing algorithms and investigated sub-optimal preemption algorithms with random selection of connections to be rerouted. We have also performed numerical and simulation comparisons of re-routing algorithms by analyzing their performance on a single link, dynamic setting and in a full network environment with a heterogeneous traffic mix.

In order to account for the users application type, QoS requirements and quantify users' value we have introduced a utility-based QoS model. The utility could be some measure of the perceived connection's quality, the user satisfaction, or even the amount paid by the user for the bandwidth allotted to it, and could provide a framework to differentiate among users on the basis of their requirements and/or revenues. Bandwidth can be then allocated such that the sum of the user utilities is maximized, subject to the link capacity constraints [32]. Our scheme allows a service provider to differentiate between different types of customers based on their priority or the service charges that they pay, in order to offer real time services, to provide QoS guarantees for multimedia traffic and to guarantee stability even in overloaded conditions.

Utility based resource allocation has recently received attention both in the wired Internet [32, 15, 12, 35] and in wireless networks [9, 50, 48]. Most of the previous work has investigated rate control algorithms based on the utilities of the users, in order to achieve the system optimal rates in the sense of maximizing aggregate utility. Our approach differs from previous work in the respect that we have investigated network utilization, QoS observed by the customers, and revenue generation perspectives for different utility-quantified bandwidth allocation schemes.

We have presented approximate analytical tools to obtain blocking probabilities in a multi rate multi class system, where users of the same class can have different resource requirements. In order to calculate the blocking probabilities for various bandwidth allocation policies, we have expanded existing multi rate Erlang B and $M/M/c/c$ models. We have considered generalizations of loss networks allowing the Complete Sharing and Guaranteed Minimum sharing policies as well as a prioritized preemptive policy. We have evaluated the blocking probabilities for a single link case and validated our approach through the simulation of such a system. Also we have expanded our single link model to calculate blocking probabilities for a multihop path, when the offered traffic of each source destination pair along the path is known.

1.3 Organization of the Thesis

The outline of this thesis is as follows. This introduction is followed in Chapter 2 by a short description of the technology that forms the basis of existing and future broadband telecommunication networks many potential applications and architectures that would benefit from the use of dynamic utility-based bandwidth allocation policies. In Chapter 3 we describe the basic concepts of bandwidth allocation policies in such networks with related work. Chapter 4 outlines characteristics and specific methods for dynamic bandwidth allocation. In Chapter 5, we describe our framework for utility based dynamic traffic management, while Chapter 6 discusses analytical model for various bandwidth allocation policies in multiservice networks. In Section 7 we present the case studies we have conducted for comparing preemption algorithms and analyzing the bandwidth allocation performances.

Chapter 8 summarizes our work with conclusions and suggestions for future work. Chapter 9 lists the publications based on this thesis.

Chapter 2

Research Context

2.1 DiffServ and MPLS Networks

Dynamic bandwidth allocation becomes a more attractive strategy in a differentiated services scenario. Traffic engineering can be used as an addition to DiffServ mechanisms to improve utilization of network resources [2, 7]. It is possible to implement some priority policies for classes in the same class-type to permit preferential access to the class-type bandwidth through the use of preemption priorities. The preempted request may then be rerouted. Preemption can be used to assure that high priority requests can be always routed through relatively favorable paths within a differentiated services environment. In the DiffServ-aware Traffic Engineering approach [4] authors define the notion of preemption and preemption priority. The preemption attribute determines whether a request with a certain priority attribute can preempt another request with a lower priority attribute from a given path, when there is a competition for available resources.

Preemption and assigning priorities to traffic trunks are motivated by the requirements for traffic engineering over MPLS. Priority and preemption express certain binary relations between traffic trunks, which determine the manner in which traffic trunks interact with each other as they compete for network resources during path establishment and path maintenance. The priority attribute defines the relative importance or "value" of traffic trunks. The priority attribute is used in constrain-based routing framework to determine the order in which path selection is done for traffic trunks at connection establishment and under fault scenarios. Priorities can also be used to impose a partial order on the set of traffic trunks according to which preemptive policies can be actualized. A protection priority could be used as a differentiating mechanism for premium services that require high reliability and a desired level of QoS to be maintained throughout the session without interruptions, such as Virtual Leased Line services, high priority voice and video traffic [3, 4].

Preemption is not considered a mandatory attribute under the current best effort Internet service model. However, in a differentiated services scenario, preemption becomes a more attractive strategy. As some protection and restoration functions may be migrated from the optical layer to data network elements to reduce costs, preemptive strategies can be used to reduce the restoration time for high priority traffic trunks under a fault conditions. In the DiffServ-aware Traffic Engineering approach [22] preemption is considered as an important piece on the bandwidth reservation and management. No preemption strategy is defined, only general guidelines are presented. The authors present scenarios where preemption can be used to:

- Limit proportion of classes on a link
- Maintain relative proportion of traffic classes
- Guarantee bandwidth services

Preemption is optional in the DiffServ-aware Traffic Engineering environment. A network administrator preferring not to use it for user traffic should be able to disable the preemption mechanisms [22].

Both RSVP-TE [2] and CR-LDP [28] provide capabilities of path preemption in similar fashion. A LSP which gets preempted must be torn down at preemption time. The preempted LSP may then be reestablished, which involves recomputing a path by Constraint Based Routing based on updated available bandwidth information and then signaling for LSP establishment along the new path. There may be cases where the preempted LSP cannot be reestablished (e.g., no possible path satisfying LSP bandwidth constraints as well as other constraints). In such cases, the network might periodically attempt to reestablish the LSP, relax the LSP constraints, or use other methods.

2.2 Wireless Networks

As new wireless systems continue to evolve to complement and replace current wireless systems, there is a distinct shift in design criteria. Future wireless systems will deal with data and will be able to support different multimedia applications. The massive use of mobile telephony and Internet services in the last couple of years, are producing extremely high expectations for the commercial success of wireless Internet access services. Future generations of cellular networks are evolving towards an all-IP architecture. Third generation systems (e.g., UMTS) will support mobile multimedia, personal services and present the convergence of digitalization, mobility, and the Internet. Integration of cellular networks and Internet is the business opportunity of the 21st century.

The next generation Internet architecture will support applications with different quality of service requirements, independently of whether users are fixed or mobile. Enabling QoS in Internet is a tough challenge, and it gets even tougher when one is introducing QoS in an environment of mobile hosts, wireless networks, and different access technologies, because of wireless networks dynamically changing topologies and resources. The rapid growth of mobile systems indicates that the future Internet will have to deal with mobile users that will use the same diversity of applications as fixed users. Thus, solutions for enabling QoS over IP should take into account mobility issues also, in order to be able to fulfil these upcoming requirements of future Internet users [18].

When designing wireless systems, the challenge is to determine the total cell capacity that should be able to satisfy the traffic load of each offered service. The total cell capacity should be partitioned among various offered services or classes of users in such a way that the overall throughput or revenue is maximized, while ensuring that the QoS constraints are satisfied.

Deterministic service guarantees and bandwidth allocation, used in IP networks, is inadequate in wireless networks with heterogeneous classes of users. A more flexible service model which allows variable QoS is needed. As cellular networks continue to evolve and offer additional services to users, reserving bandwidth exclusively and statically for each type of service, under the fluctuating traffic load and variable wireless channels, may not be efficient from network utilization and revenue generation perspectives. Priority-based access control and service policies may be an attractive alternative to a multiservice cellular network. Priority based schemes could be used to optimize the blocking probability and increase network utilization, by fine tuning the bandwidth available to types of services, thus reducing the impact of fluctuations in the arrival rate and variations in channel capacity. The priority based schemes achieve better gains in throughput and resource utilization for higher priority services. Instead of splitting resources for services, preempting scheme decreases dropping of high priority calls as the rate of lower priority calls increases. The idea of preemption is to offer resources in a more dynamic way. Resources are shared among different services, but when the usage of one service exceeds the available resources, preemption allows the traffic with higher priority to seize the resources of other traffic to maintain service quality.

User priorities can be assigned based on:

- the amount of resources requested/type of service.
- the number of networks a user can access at the location: the higher the number of networks a user can access, the lower is the priority of the user, as he has higher chances to be admitted in another network if later preempted than a user who can access only one network.
- the user service level agreement (SLA).
- the mobility pattern - a faster moving user is more likely to free up the network resources in a shorter time than a slower user.

Another aspect of wireless networks is that they are becoming more prevalent in protecting lives and responding to emergencies. Heavy traffic demands are placed upon cellular networks after the disaster occurs, causing severe network congestion and high call blocking in communications. The wireless priority access service could allow designated military and civilian personnel, as well as state and local emergency crews to access cell channels ahead of the general public in times of crisis. In the next section we present the use of priority access in providing Quality of Service assurance and network survivability and reliability through Traffic Engineering.

2.3 Network Survivability and Reliability Through Traffic Engineering

E-commerce and mission-critical Internet services require a maximum of availability from the network and a minimum of network outage times. The current Internet has a built-in degree of survivability due to the connectionless IP protocol. Dynamic routing protocols react to faults by changing routes when routers learn about topology changes via routing information updates (e.g., link status advertisements). Loss of QoS has not been an issue because current Internet traffic is best-effort. The new connection-oriented, real-time interactive services that are already being offered on the Internet (or are currently emerging) have increased resilience requirements. Quality of Service assurance is becoming a necessity in the Internet of future. Traffic-engineering methods that allow the provisioning of network resilience are a clear requirement for current and future Internet networks.

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults, by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Due to the increasing demands to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet network survivability represents a requirement for the future networks [4].

During major disaster events, the public network experiences severe stress due to damaged infrastructure and heavy traffic loads. As bandwidth becomes severely constrained, it becomes difficult to establish and maintain communication sessions. It is essential that disaster recovery operations be able to readily communicate and be given a high probability of successful completion to organize and coordinate essential activities. The telecommunication capabilities for handling authorized emergency traffic should be accomplished using existing applications and standards whenever possible with any necessary adjustments needed to support preferential treatment of emergency traffic during severe periods of congestion [5].

Many methods have been proposed for QoS assurance in case of network failures or load fluctuations. Packet switching and TCP/IP as the technological foundations of the Internet, do not guarantee survivable communications in the event of an attack at a node or a link. In case of failure a large number of connections would require a simultaneous retransmission of lost packets. Retransmitted packets create a backlog at the traffic source that combined with the collective attempt to reestablish connections may cause undesirable transients and congestion in the network, which is the dominant factor on network performance immediately after a failure. If this transient congestion is not controlled it can spread and lead to a network breakdown.

One way of improving network dependability is to prepare *redundant network resources* in order to cope with failures or load fluctuations. In self-healing algorithms such spare capacity is calculated and assigned in the design phase. When failures occur, this pre-assigned reserved capacity is used to restore the failed connections.

Another approach toward improving network dependability involves the use of *reconfigurable networks* [24] and *self-sizing networks* [40]. Reconfigurable network is a network where the effective topology and capacities can be dynamically adapted to changes in traffic requirements or to changes in the physical network due to failures. Self-sizing network operation is a traffic engineering and operation concept developed for ATM networks which allows networks to be rapidly operated and re-dimensioned, based on measurement of traffic flow and demand.

Reconfigurable networks and self-sizing network restore connections after failures or load fluctuations using centralized reallocation of resources from working connections to degraded connections. These centralized algorithms are not suitable for restoration after failures. It is difficult to achieve rapid restoration using a centralized approach, especially in a large networks. Therefore, an algorithm for resource reallocation must be decentralized [20].

2.4 Chapter Summary

There are many potential applications and architectures that would benefit from the use of dynamic utility-based bandwidth allocation policies. This chapter was meant to present three of them which we consider the most important. First section describes utility-based bandwidth allocation policies in DiffServ-aware and MPLS networks. Second section describes utility-based bandwidth allocation policies in wireless environment.In third section we outline the use of utility-based bandwidth allocation policies in providing Quality of Service assurance and network survivability and reliability.

Chapter 3

Static Bandwidth Allocation Policies

In the most general case of resource allocation, called a *complete sharing* (CS) admission policy, all connections are admitted if resources are available at the time a connection is requested. In this case the only constraint is the overall system capacity. Naturally, in a CS policy, connections that request lower amount of bandwidth are more likely to be admitted. This policy does not consider the importance of a connection when allocating resources to a new request.

3.1 Optimal Bandwidth Allocation Policies

The Optimal Access Problem : In [19] and [29] authors consider the problem of optimal dynamic allocation of a limited number of processors to service the demands from users with different rates at which they enter requests for service and the mean length of time occupy a single processor. G.J. Foschini et al. formulate the mathematical model for any finite number of customer types and provide solution for the case of two competing customer classes [19]. In [29] the authors prove a conjecture for the optimal policy for a system introduced in [19]. Ross and Tsang in [47] consider the stochastic knapsack problem motivated by the problem of accepting and blocking the calls to a circuit-switched telecommunications system which supports a variety of traffic types. The classical knapsack problem is to pack a knapsack of integer volume F with objects from K different classes in order to maximize profit. The authors discuss the optimal control policy in order to accept/reject the arriving objects as a function of the current state where for each class a portion of knapsack is dedicated. B. Kraimeche and M. Schwartz [34] introduce a class of restricted access control strategies capable of providing the improved system performance for integrated communications system handling traffic demands of heterogeneous users. These access strategies consist of grouping the set of user types and limiting the amount of bandwidth occupied by user types in each group. The optimum design of a restricted access strategy is a combinatorial optimization problem. In a paper by Gopal and Stern [25], the authors formulate the problem of controlling the access to a communication channel in an integrated service environment with traffic demands of heterogeneous users as a Markovian Decision Process and propose the Policy Iteration algorithm to find an optimal call blocking policy and sub-optimal policies which can be found with less computational effort.

The Optimal Multi-class Admission Problem : Consider a single network with n classes of traffic and total available bandwidth B. Each traffic class $i \in \{1, ..., n\}$ consists of a stream of calls with Poisson arrivals at rate λ^i and holding times μ^i. The admission control will decide whether arriving calls should be admitted to the network or whether they should be blocked. The system is modelled as a Markov chain.

Let $\mathbf{x} \equiv \{\mathrm{x_1, ..., x_n}\}$ represent the state of the system where $x_i \equiv$ number of type i requests processed. A is a $m \times n$ matrix, with column i indicating the number of each of m resources used by service i. Z represents the state space, $Z \equiv Ax$.

The transition rates of the of this Markov chain are given by:

$$r_{xy} = \begin{cases} \lambda_i, & if \quad y = (x_1, ..., x_i + 1, ..., x_n) \\ x_i \mu_i, & if \quad y = (x_1, ..., x_i - 1, ..., x_n) \\ 0, & else \end{cases}$$

Let $\pi(x)$ denote the equilibrium probabilities of the Markov chain, where

$$\pi(x) = \pi(0) \textstyle\prod_{i=1}^{n} \frac{\rho_i^{x_i}}{x_i!}$$

$$\pi(0) = \frac{1}{\sum_{x \in Z} \prod_{i=1}^{n} \frac{\rho_i^{x_i}}{x_i!}}$$

The performance criterion for the optimality of the control can be: System throughput or bandwidth utilization (from the network administrator point of view) or call blocking probabilities (from the network user point of view). The two performance criteria can be seen as equivalent for the purpose of admission control since they are linearly related [29].

Let the row vector $B \equiv \{B_1, ..., B_n\}$ represent the amount of bandwidth occupied by each call of class i.

Average system throughput: Let I be the global resource occupancy $I = \sum_{i=1}^{n} x_i B_i$ The equilibrium probabilities $q(i) = P_r[I = i]$ of the I are obtained by writing $q(i) = \sum_{\mathbf{x}\cdot\mathbf{B}=\mathrm{i}} \pi(\mathbf{x})$. The average system throughput is then

$$E(I) = \sum_{i=0}^{B} iq(i)$$

3.2 Sub-Optimal Bandwidth Allocation Policies

In a *complete partitioning* (CP) policy, every class of traffic is allocated an amount of bandwidth that can only be used by that class.

A *trunk reservation* (TR) control policies provide a simple means of deciding when calls should be accepted: a decision at time t depends only on the free capacity of each resource of a network at time t, and not on a detailed description of the entire state of the network. Further, such policies provide a very robust control mechanism: given only that trunk reservation parameters are chosen within some sensible range, the behavior observed is typically rather insensitive to call arrival rates. Suppose that the finite set $\mathbf{I}$ indexes the types of call offered to the resource. A trunk reservation policy says that class i may use resources in a network up until the point that only r_i units remain unused. An accepted call of type i earns reward at a rate of ω_i per unit time in progress. If call types are labelled so that $\omega_1 \geq \omega_2 \geq \ldots \omega_I$, where $I =\mid \mathbf{I} \mid$, then $r_1 \leq r_2 \leq \ldots \leq r_I$ and $r_1 = 1$ (since an optimal policy will clearly accept some calls when one or more units of capacity are free). The optimal choices of trunk reservation parameters $r_i, i \in \mathbf{I}$, can be determined from the stationary distribution of the birth-death process describing the number of units of capacity in use, or via policy improvement. Optimal values of trunk reservation parameters are studied in detail in [33], [45], where various limiting regimes are considered. Lippman [37] has shown that the same form of policy is optimal under a variety of discount and finite-horizon criteria. Also, Nguyen [41] considers a closely related problem with $I = 2$ but where one arrival stream is Poisson and the other is an overflow stream from an $M/M/m/m$ queue: again the optimal policy is of the (generalized) trunk reservation form [36].

A *guaranteed minimum* (GM) policy gives each class their own small partition of resources to guarantee that there is always space for a certain number of requests. Once used up, classes can attempt to use resources from a shared pool of resources. An *upper limit* (UL) policy places an upper limit on the number of requests from each class that can be in service at any time. These policies provide what can be called *significant separation with sharing*, and outperform complete partitioning and complete sharing. Variations on the UL and GM bounds also exist. Alternative bounds can be based on other linear constraints on the number of requests from each class. These alternative bounds include UL bounds for classes of customers. Such additional constraints may increase the computational complexity of the algorithm.

The UL and GM bounds have been specifically proposed in a special context by Kamoun and Kleinrock [30]. The UL and GM bounds lead to a coordinate-convex sharing policy, so that the resulting resource-sharing model has a product-form steady-state distribution. The UL bounds can easily be enforced if the resource provider keeps track of the number of requests from each customer in service, the GM bounds are more difficult to enforce. The UL and GM bounds may either be provided directly by the customer or determined by the resource provider to satisfy other requirements. The customer must provide its blocking requirement and a characterization of its desired traffic. The request blocking requirement is the maximum allowed request blocking probability, assuming that all customers generate requests according to their negotiated traffic parameters [11]. Optimal values of the guaranteed minimum and upper limit parameters for calculating the blocking parameters and other steady-state descriptions are studied in detail in [10] as multi-dimensional generalizations of the classical Erlang loss model.

The CP, GM, UL, and TR policies were found to outperform the CS policy when significant differences between classes exist in terms of bandwidth and offered load. UL and GM policies also significantly outperform TR policies using controlling blocking parameters to determine an effective response in the presence of temporary overloads or resource failure [10].

The above policies are effective when network traffic behaves consistently with the initial assumptions about traffic characteristics made to implement the policies. For those reasons, policies that are robust when class loading increases beyond engineered loading have been developed. Virtual partitioning (VP) [8] uses a variant of trunk reservation, where classes are assigned one trunk reservation level but if the current capacity usage by a class exceeds its nominal allocation at a certain point in time, it is given a lower priority in the admission of new requests. The objective of the virtual partitioning scheme is to prevent heavily loaded classes from degrading the performance of those that have loads within their prescribed bounds. Overloaded classes that already experience high blocking from being overloaded are penalized further by having more restrictive trunk reservation imposed [8].

The major drawback of guaranteed bandwidth policies is inefficiency; the bandwidth received may not actually be fully utilized. Resources are wasted when high priority users are not offering much traffic to the network, because the capacity reserved for them goes unused.

To avoid these disadvantages, preemption can be used. The concept is to allow all users into the network whenever resources are available, then interrupt and discontinue the flows of lower priority users if high priority users need resources but there is no room to accommodate them. Connection preemption, when coupled with the capability to reroute connections provides a high quality of service to higher-priority network connections. It can utilize the bandwidth more efficiently in case of failure or load fluctuation by allowing the higher priority connections to get through in order to achieve a desired level of network availability. Disrupting a connection has obvious disadvantages: all the work that was done so far may be lost, and packets would need to be retransmitted. However, the ability to disrupt certain types of connections and perform a reroute based on the current state of the network as a policy can greatly improve the performance of the network, let alone allow crucial or emergency services to be performed uninterrupted.

3.3 Chapter Summary

A brief overview of mechanisms of static bandwidth allocation policies are presented in this chapter. The most general case of resource allocation, is the complete sharing admission policy, in which all connections are admitted if resources are available at the time a connection is requested. Optimal policies take the state space (allowable combinations of numbers of connections from each class) from complete sharing policy and constrain it in some way. In order to implement optimal policies a detailed accounting of every allowable network state and state transition is necessary, which is impractical for networks of even modest size. Therefore, a set of sub-optimal policies have been developed that we present in the second section, which are simpler to implement and provide a more intuitive understanding of how resources are managed.

Chapter 4

Dynamic Bandwidth Allocation

4.1 Distributed Solution Approach

We present distributed algorithms that address how one might *enhance*, if not optimize, average user-perceived performance and describe how it can be implemented in a real network. If the system has information about all users, the optimization problem of maximizing the sum of the user values may be mathematically tractable. However, in practice not only is the system not likely to know the information about all users in the network, but also it is impractical for a single centralized system to compute and allocate the users' rates, due to the computational intractability of the problem for large networks. The increasing complexity and size of the Internet make centralized bandwidth allocation policies impractical. We study the feasibility of achieving the maximum total value of the users in a distributed environment, using only the information available at the *end hosts*.

Priority-based bandwidth allocation algorithms, with low time complexity, can be used to optimize the blocking probability and increase network utilization, by offering resources in a *more dynamic way* and providing preferential treatment for some services. The importance or "value" of a connection can be expressed by a priority level. The priority levels can be preassigned by the end-system or by the network administrator based on the type of the user. Alternatively, this can be done by using various factors, in order to relate to the connections Quality of Service (QoS) requirements, such as reliability of packet delivery, pricing, bandwidth requirement, timeliness, desired blocking probability, duration of the connection, geographical distance, or nature of the traffic such as voice, video, or WWW traffic.

Once a request for a new connection arrives, the routers on the path to be established by the new request need to check for bandwidth availability on all links that compose the path. For the links in which not enough bandwidth is available, an algorithm has to decide which ongoing connections of lower priority to preemp to be rerouted in order to establish the high-priority connection. The algorithm is run *locally* in each link in order to guarantee the end-to-end bandwidth reservation for the new request. This is a decentralized approach, in which every node on the path would be responsible to independently determine which connections would be rerouted in order to accept the new request. For these reasons, a decentralized approach, although easier to be integrated in the current Internet environment, may not lead to a strictly optimal solution.

4.2 Dynamic Resource Allocation Algorithms

Much recent work has been accomplished to formulate optimal and approximate on-line algorithms for finding the best combination of which connections would be rerouted, if high priority users need resources but there is no room to accommodate them. One of the crucial performance objectives of these algorithms is that they can be deployed over the Internet without significant modification within the network.

In this framework, priority-based bandwidth allocation algorithms with low time complexity can be used to optimize the blocking probability and increase network utilization, by offering resources in a more *dynamic* and by providing preferential treatment for some services. The concept is to allow all users into the network whenever resources are available, interrupt and perform a reroute of the flows of less "valuable" users based on the current state of the network, if more "valuable" need resources but there is no room to accommodate them.

4.2.1 Optimal Algorithms

The two parameters describing a connection are, namely, *bandwidth* and *priority*. Which connections would be preempted to be rerouted, if high priority users need resources but there is no room to accommodate them, can be determined by optimizing an objective function over these two parameters of the connections, and the number of connections to be rerouted. The objectives could be any or a combination of the following:

1. Reroute the least amount of bandwidth. Network bandwidth is better utilized and there is minimum disruption of user traffic.

2. Reroute the connections that have the least priority. There is less disturbance of high-priority connections and the QoS of higher priority users is better satisfied.

3. Reroute the least number of connections. A minimum number of connections have to

be rerouted.

4. Reroute the traffic according to the a performance criteria for each traffic type. Traffic should be rerouted as much when necessary, not whenever possible.

In [43] the authors proposed two algorithms they named *Min_BW* and *Min_Conn* that optimize the criteria above in a certain order of importance. The algorithm *Min_BW* optimizes the criteria of (i) the amount of bandwidth to be rerouted, (ii) the priority of connections to be rerouted, and (iii) the number of connections to be rerouted, in that order. The algorithm *Min_Conn* optimizes the criteria of (i) the number of connections to be rerouted, (ii) the bandwidth to be rerouted, and (iii) the priority of connections to be rerouted, in that particular order. In [13] the authors proposed an objective function that can be adjusted by the service provider in order to stress the desired criteria for optimization and derive a heuristic which approximates the optimal result. These algorithms are globally and strictly optimal with respect to their objective functions because they perform an *exhaustive* search to select a solution based on the criteria.

4.2.2 Sub-optimal Lottery Algorithm

In [53, 54] we have proposed and analyzed a *probabilistic selection* of connections to be rerouted from the set of connections with lower priorities and concluded that probabilistic selection algorithms could provide a high quality of service to higher-priority network connections, while utilizing network bandwidth efficiently.

In this approach we do not aim to explicitly optimize any of the objectives listed above. Our goal is to *almost instantly* find a set of connections to be preempted in order to accommodate the preempting connection. This is very important in the case of a large network with a large number of connections, given that connection preemption will need to be implemented as a real-time procedure. The resulting sub-optimal policy, due to its *linear complexity* reduces dramatically the time to find a set of connections to be preempted while maintaining a very high level of network resources utilization and a minimum disturbance of high priority connections.

An upper bound on the computational complexity of our sub-optimal algorithms is $O(k)$ where k is the number of connections using the link and having a priority less than that of the preempting connection. This is more favorable than *Min_BW* with $O(k \cdot 2^k)$ computational complexity or *Min_Conn* with $O(k^2)$ computational complexity in case of a large network with a large number of connections. We compare all the algorithms listed above in case studies presented in Chapter 7.

4.3 Chapter Summary

In this chapter we have discussed how dynamic bandwidth re-allocation can be used to provide the bandwidth to a connection just for the time it is actually requested and if necessary adjust the allocated bandwidth according to the condition of the network, thus improving the flexibility of the network. Solution must be decentralized, to be easier to be integrated in the current Internet environment. Priority-based bandwidth allocation algorithms, with low time complexity, can be used to optimize the blocking probability and increase network utilization, by offering resources in a *more dynamic way* and providing preferential treatment for some services. Every node on the path would be responsible to independently determine which connections would be rerouted in order to accept the new request.

Chapter 5

Utility-based QoS Model

Dynamic bandwidth re-allocation can be used to provide bandwidth to a connection just for the time it is actually requested and, if necessary, to adjust the allocated bandwidth according to the condition of the network, thus improving the overall flexibility. Our framework for utility based QoS model is described in the following:

- **Applications** :

  We focus on *four* types of applications which can be mapped to the QoS architecture of wireless networks as well as DiffServ standards [7, 39]. The first type of applications that we consider are called *hard real-time applications*, which need their data to arrive within a given deterministic delay bound. Examples are disaster recovery and emergency traffic or some important advanced applications, such as remote surgery or remote instrument control. The second type are *delay-adaptive applications* which are more tolerant of occasional delay bound violations and dropped packets. The third type are the *rate-adaptive applications* which can adjust their transmission rate in response to network congestion. The fourth type are *elastic applications*, which have more relaxed or lower quality of service requirements [49].

- **Constraints** :

The Quality of Service is a very comprehensive concept which may involve many factors in analyzing different parameters. Emerging high-speed networks will be required to offer a set of classes of service (CoSs), e.g., Diffserv and provisioned LSPs in MPLS networks, directed to a diverse set of users. In addition, they will need to strike more complex and resource-specific service-level agreements (SLAs) in order to satisfy the needs of the next generation of applications. Each application will define its QoS requirements along d dimensions $\{Q_1, Q_2, \ldots, Q_d\}, d \geq 1$. An application is feasible if its QoS constraint is satisfied in every QoS dimension.

We consider three constraints on the network users' QoS requirements, namely throughput, loss and timely delivery in order to investigate the restoration time required when failures or load fluctuations occur. During restoration applications are disrupted. How long an acceptable disruption period is, varies with the type of application. Some applications need restoration in less than a second. Typical examples are real-time systems for air traffic control, military systems that depend upon uninterrupted service, and time critical applications used within telemedicine. Other applications such as routine telephony and data transactions can tolerate outage periods of several seconds. Another group of applications is the one used largely for entertainment purposes which can tolerate outage times in the order of minutes.

The constraint on the system is that the sum of resources R_i used by each of the

n applications cannot be larger than the total capacity of the resource R, that is, $\sum_{i=1}^{n} R_i \leq R$. In case of failure or load fluctuation, a network does not have enough resources to allocate in order to guarantee the QoS for every user, which affects the traffic handling capacity of the system. The traffic handling capacity represents network utilization and the ratio of the number of restored connections to the number of connections affected by failure or fluctuations in the network.

For our experiments we will describe the service in terms of the amount of its effective bandwidth. A request r_k can be defined as a flow of information from a source to a destination involving a certain amount of "bandwidth". Bandwidth can be represented by the peak rate of the request R_k and the minimum rate of the connection m_k, duration. The allocated bandwidth, $\hat{c}_k$, for the request r_k can then be calculated as the *effective bandwidth* according to the model introduced in [27], [1].[1]

- **Utility functions** :

  For each type of application, we define a specific utility function $u(b)$ which represents the "level of guarantee" provided to a user by the network or "insurance" in case of transient overloads or network faults. A utility function describes how the performance of an application changes with the amount of effective bandwidth it receives. In our model, each user (or aggregate of users, to maintain scalability) signals its application type and QoS requirements. Applications that are designated to be transmitted at a fixed rate with no interruptions may generate a request that has a *step* utility function

[1] The reader should note that, without loss of generality, we utilize statistical effective bandwidths in our generalized framework. However, several alternative quantities of similar use, for example deterministic effective bandwidths or traffic envelopes can be used instead [44, 23, 17].

as shown in Figure 5.1(a). In case of a step utility function, the received bandwidth is equal to the peak rate of the request. Applications that are designated to *adapt* to transmissions and transmit at a variable rate, delay adaptive and rate adaptive applications, may generate a request that has a utility function, as shown in Figures 5.1(b) and 5.1(c), respectively. Elastic applications generate requests with a utility function such as shown in Figure 5.1(d) [49]. The utility function must be concave for the bandwidth values each application can take.

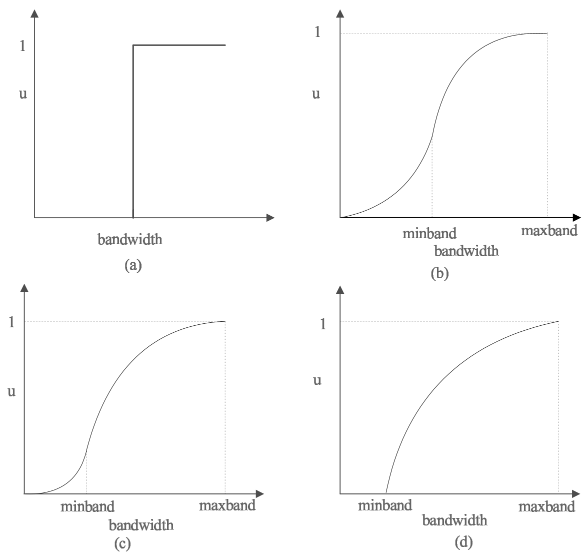

Figure 5.1: Utility (performance) of hard real-time application (a), delay adaptive application (b), rate-adaptive application (c) and elastic application (d) as a function of bandwidth.

The priority level is assigned by quantizing the *weighted utility* of the request into N levels, where N cannot be too large or too small. In the MPLS model [4], [22] priorities can take values in the range from zero (0) to seven (7), with the value zero (0) being the priority assigned to the most important path. Therefore, the range of priority values is divided (without loss of generality) into eight levels using the quantizing function $q(u)$ in

Figure 5.2.

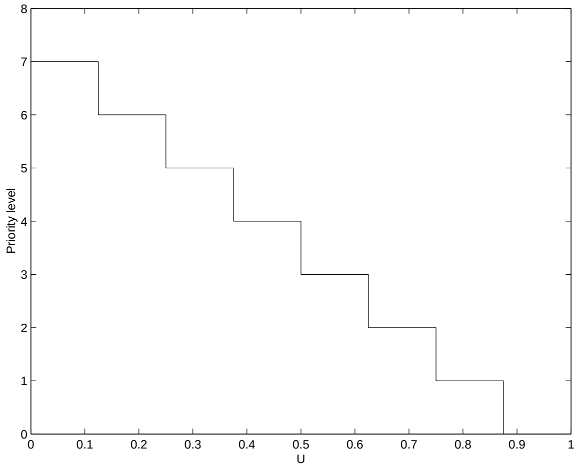

Figure 5.2: Quantizing function for assigning priorities based on the weighted utility of the request.

We consider a network with a set of nodes N, set of resources or links J where link j has a capacity C_j and set of users K. Each user has a fixed route J_k, which is a nonempty subset of J. Let K_j represent a set of requests who share the link $j \in J$. Let S represent the set of satisfied requests that arrive in the network over a certain period of time. The problem that we are trying to solve is to determine user bandwidth allocations c_k under the following objective and constraints. The total utility of all requests in S which we want to maximize is the sum of all utilities:

$$\max \sum_{k \in S} u_k$$

assuming that the utilities are additive, which is a reasonable assumption since the network users are independent, subject to the constraints:

$$\sum_{k \in K_j} c_k \leq C_j, \quad \forall j \in J$$

indicating that the total rate of the sessions using a link cannot exceed the capacity of the link.

Utility based resource allocation has recently received attention both in the wired Internet [32, 15, 12, 35] and in wireless networks [9, 50, 48]. Most of the previous work has investigated rate control algorithms based on the utilities of the users while being fair, in order to achieve the system optimal rates in the sense of maximizing aggregate utility. Our approach differs from previous work that we investigate network utilization, QoS observed by the customers and revenue generation perspectives for different utility described bandwidth allocation schemes.

5.1 Chapter Summary

In this chapter, we have introduced a utility-based QoS model and a utility-based bandwidth allocation scheme which accounts for the users QoS requirements and actively adapts to the dynamics of the network. The utility of a request is an arbitrary function of the bandwidth received during its session, depending upon the application generating the request. In our model each user (or aggregate of users, to maintain scalability) signals its application type and QoS requirements. Bandwidth can be then allocated such that the sum of the user utilities is maximized, subject to the link capacity constraints.

Chapter 6

Analytical Bandwidth Allocation Model

6.1 Bandwidth Allocation Policies

We describe the QoS observed by the customers in terms of a customer's request *blocking probability*, the proportion of requests which either could not be admitted into the system or could not be completed once admitted because of being preempted by the arrival of a higher priority connection. In case of high blocking probabilities, the network fails to satisfy customer requirements, but in case of low blocking probabilities, network resources are not used efficiently. Bandwidth allocation policies are used to determine admissibility of a given set of customers with their resource requirements for a given resource capacity.

The nature of the bandwidth allocation problem is such that a decision made previously to accept a connection may have been wrong because it caused a future more valuable connection request to be rejected. For example, an accepted connection may have used up all the capacity of the network, thereby causing the connection of much larger bandwidth requirement (consequently larger revenue to the network) to be rejected.

We consider the case with R traffic types, where each traffic type requires a number of resource units, which does not have to be the same for different requests of the same traffic type. Traffic types are described by their arrival rate λ_r, mean service time $1/\mu_r$, $A_r = \lambda_r/\mu_r$ is the offered traffic. The number of resource units for each connection of a traffic type r is distributed according to a known distribution $f_r(b)$ and they all share a common number of servers C.

In the most general case of resource allocation, called a *complete sharing* (CS) admission policy, all connections are admitted if resources are available at the time a connection is requested. In this case the only constraint on the system is the overall system capacity. Naturally, in a CS policy, connections that request lower amounts of bandwidth are more likely to be admitted (e.g., a voice connection will more likely be admitted compared to a video connection). This policy does not consider the importance or value of a connection when allocating resources.

However, it is often desirable to consider sharing policies which provide different grades of service or protect one traffic class from another. These policies can be used to provide preferential treatment for some services, in order to handle authorized emergency traffic or due to revenue considerations. Guaranteeing QoS to the requests means that user can explicitly specify certain requirements for a request such as throughput, loss or delay, when establishing a so-called Service Level Agreement (SLA) with an Internet Service

Provider or a network operator, a contract defining clear responsibilities for the operator with precise specification of the kind of network service required from the customer [4].

We analyze *the guaranteed-minimum* sharing policy, which reserves a specified number of circuits for each traffic class [11] and *a preemptive policy* in a prioritized multi class system, in which connections of lower priority can be interrupted, if high priority users need resources but there is no room to accommodate them [59], [43], [13], [54]. Utility-based preemptive bandwidth allocation algorithms with low time complexity can be used to optimize the blocking probability and increase network utilization, by offering resources in a more *dynamic* and by providing preferential treatment for some services. The concept is to allow all users into the network whenever resources are available, interrupt and perform a reroute of the flows of less "valuable" users based on the current state of the network, if more valuable users need resources but there is no room to accommodate them. Disrupting a connection has obvious disadvantages: all the work that was done so far may be lost, and packets would need to be retransmitted. However, the ability to disrupt certain types of connections and perform a reroute based on the current state of the network as a policy can *greatly improve* the performance of the network as a whole, let alone allow crucial or emergency services to be performed uninterrupted.

The idea of such general models is not new: Kaufman [31] and Roberts [46] presented recursive algorithms for computing the blocking probabilities for a multirate Erlang-B model. Choudhury, Leung, and Whitt [10], and Nilsson [42], presented stable and efficient algorithms for calculating the blocking probabilities for multi-rate multi class systems. Optimal values of guaranteed minimum for calculating the blocking parameters and other steady-state descriptions are studied in detail in [10], as multi-dimensional generalizations of the classical Erlang loss model. On the other hand, very little work has been done in analytical modelling preemptive policies and examining the merit of preemptive versus non-preemptive approaches [34], [6], [60].

6.1.1 Complete Sharing Policy

Let the steady state vector for the number of subscribers n_r of each type r in the system be $\pi(\underline{n})$, where $\underline{n} = (n_1, n_2, \ldots, n_R)$. The state space consists of all allowable states such that $\sum_{r=1}^{R} n_r b_r \leq C$.

Since arrivals are assumed to be Poisson, the probability of blocking for class r, denoted as B_r, is the probability of being in any state vector in the state space such that not enough resources are available to support the arrival of another connection of type r.

The steady state probability mass function has the product form

$$\pi(\underline{n}) = \frac{1}{G} \prod_{r=1}^{R} \frac{A_r^{n_r}}{n_r!}$$

where G is the normalizing constant such that the probabilities add up to one.

$$G = \sum_{0 \leq \sum_{r=1}^{R} n_r b_r \leq C} \prod_{r=1}^{R} \frac{A_r^{n_r}}{n_r!}$$

To calculate the blocking priority for the complete sharing policy we apply the approach defined in [42]. Blocking probabilities are computed from

$$B_r = \sum_{j=0}^{b_r - 1} \beta(n, j)$$

where $\beta(m,k)$ is defined recursively as

$$\beta(m,k) = \frac{\beta(m-1,k-1)}{1+\frac{1}{m}\sum_{r=1}^{R} A_r b_r \beta(m-1,b_r-1)}$$

$$\beta(m,0) = \frac{\frac{1}{m}\sum_{r=1}^{R} A_r b_r \beta(m-1,b_r-1)}{1+\frac{1}{m}\sum_{r=1}^{R} A_r b_r \beta(m-1,b_r-1)}$$

$$\beta(0,0) = 1$$

where $0 \leq m \leq C$ and $0 \leq k \leq m$.

The number of calls of a traffic type r in the system is obtained from Little's formula

$$N_r = A_r(1-B_r)$$

The total utility of the customers in the system is

$$U = \sum_{r=1}^{R} N_r \cdot u_r(b_r)$$

The blocking probability B_r, number of calls N_r, and total utility of customers U are functions of resource requirements of all traffic types.

To find the average blocking probability and the average total utility we need to find an average value over all the values that $b_1, \ldots, b_R$ can take:

$$pb_r = \int \ldots \int B_r(b_1, \ldots, b_R) f(b_1, \ldots, b_R) db_1 \ldots db_R \tag{6.1}$$

$$E[U] = \int \ldots \int U(b_1, \ldots, b_R) f(b_1, \ldots, b_R) db_1 \ldots db_R \tag{6.2}$$

In order to calculate blocking probabilities, classes must be defined such that

$$b_1 \leq b_2 \leq \ldots \leq b_R$$

When evaluating blocking probabilities of each traffic type for all values $b_1, \ldots, b_R$ can take, we must first sort the traffic types according to the amount of requested resource units in increasing order, and then calculate their blocking probabilities.

6.1.2 Guaranteed Minimum

For the guaranteed minimum policy, each application type r is assigned its guaranteed minimum of resources C_r. The connection first tries to be established within the guaranteed minimum resources. Connections which are blocked in the guaranteed minimum of the resources, compete with connections of other traffic types, which are blocked in their guaranteed minimum of the resources, as in complete sharing policy in the shared part of resources, $C - \sum_{r=1}^{R} C_r$.

For each traffic type r we find the blocking probability in the guaranteed minimum of the resources as for the M/M/c_r/c_r system. The quantity c_r represents a maximum number of connections of traffic type r which can be established at the same time within a guaranteed minimum of the resources C_r and is equal to

$$c_r = \lfloor \frac{C_r}{b_r} \rfloor$$

The Erlang B formula gives the probability of blocking for $M/M/c_r/c_r$ systems:

$$B'_r = E(c_r, A_r) = \frac{A_r^{c_r}/c_r}{\sum_{k=0}^{k=c_r} \frac{A_r^k}{k!}}$$

which can be computed efficiently from a recursive formula

$$E(n, A) = \frac{AE(n-1, A)}{n + AE(n-1, A)}$$

with $E(0, A) = 1$.

The blocking probability B''_r in the shared portion of the resources is found using the same recursive algorithm as for the CS policy by approximating the blocked traffic from guaranteed minimum of resources to be Poisson. The number of shared servers is $C - \sum_{r=1}^{R} C_r$, with offered traffic per traffic type now being $B'_r \cdot A_r$.

The total blocking probability is then equal to

$$B_r = B'_r \cdot B''_r$$

The number of calls of a traffic type r in the system is obtained from Little's formula

$$N_r = A_r(1 - B_r)$$

The total utility of the customers in the system is

$$U = \sum_{r=1}^{R} N_r \cdot u_r(b_r)$$

Finally, the average total blocking probability pb_r and the average total utility $E[U]$ are found using equations (6.1) and (6.2).

6.1.3 Preemptive policies

Application traffic is mapped to K priority classes. Let the A_{rk} be the portion of the offered traffic A_r of the application type r which takes priority level k.

$$A_{rk} = A_r \cdot \int_{u_r^{-1}(1-(k+1)/K)}^{u_r^{-1}(1-k/K)} f_r(b)db$$

Blocking probabilities for each priority level k are obtained recursively starting from the highest priority class 0. The maximum amount of resources each priority level k can utilize, C_k, is equal to

$$C_k = C_{k-1} - \rho_{k-1} \cdot C_{k-1}$$

with $C_0 = C$ and , where ρ_k is the system utilization for priority level k.

The blocking probability B_{rk} of each portion of traffic type r with a priority level k is obtained using the algorithm as for the CS policy with C_k number of shared servers and A_{rk} as the offered traffic of each traffic type.

The number of calls of traffic type r with a priority level k in the system is obtained from Little's formula

$$N_{rk} = A_{rk}(1 - B_{rk})$$

The total utility of customers with a priority level k in the system is

$$U_k = \sum_{r=1}^{R} N_{rk} \cdot u_r(b_r)$$

To find the application type r average blocking probability for priority level k, pb_{rk}, and the average total utility $E[U_k]$ for priority level k we need to find an average value over all the values that $b_1, \ldots, b_R$ can take for that priority level:

$$pb_{rk} = \int \ldots \int B_{rk}(b_1, \ldots, b_R) f_k(b_1, \ldots, b_R) db_1 \ldots db_R$$

$$E[U_k] = \int \ldots \int U_k(b_1, \ldots, b_R) f_k(b_1, \ldots, b_R) db_1 \ldots db_R$$

The system utilization ρ_k for priority level k is equal to

$$\rho_k = \int \ldots \int \frac{\sum_{r=1}^{R} N_{rk} b_r}{C_k} f_k(b_1, \ldots, b_R) db_1 \ldots db_R$$

The blocking probability per application type is then equal to

$$pb_r = \sum_{k=0}^{K-1} \frac{A_{rk}}{A_r} \cdot pb_{rk}$$

Finally, the total utility of customers in the system is

$$E[U] = \sum_{k=0}^{K-1} E[U_k]$$

6.2 Multihop Path Model

In this section we present how to calculate blocking probabilities for a k-hop path. A k-hop path consists of $k+1$ nodes labelled $0, 1, \ldots, k$ and hop $l, l = 0, \ldots, k-1$ represents the link between nodes l and $l+1$ as on Figure 6.1.

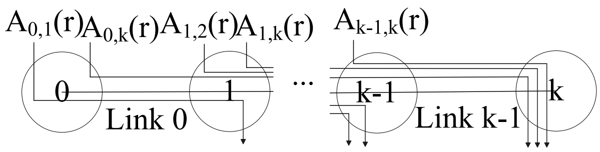

Figure 6.1: A k-hop path.

This method can be used if traffic intensity is such that, for all links, the sum of blocking probabilities of all source-destination-pair traffic on the link, if only that source-destination-pair traffic is present, retains less than 100%. For a lightly to moderately loaded path we assume that overflow traffic retains its Poisson characteristics.

We use the following notation:

- $A_{i,j}(r)$ represents the offered traffic of traffic type r with i as a source node and j as a destination node, $j > i$.

- $A^l_{i,j}(r)$ represents the portion of the offered traffic of traffic type r with i as a source node and j as a destination node on the link l.

- $A^l(r)$ represents the offered traffic from traffic type r on link l.

- $B_{i,j}(r)$ represents the blocking probability of traffic type r with i as a source node and j as a destination node, $j > i$.

- $B^l_{i,j}(r)$ represents the blocking probability of traffic type r with i as a source node and j as a destination node, $j > i$, on the link l.

- $B^l(r)$ represents the blocking probability of a traffic type r on the link l.

Our goal is to find $B_{i,j}(r)$ for each source destination pair along the path, knowing $A_{i,j}(r)$, the offered traffic of each source destination pair for all traffic types.

*Lemma:*The total $B_{i,j}(r)$ can be found as:

$$B_{i,j}(r) = \bigcup_{l=i}^{j-1} B^l_{i,j}(r) = \sum_l B^l_{i,j}(r) - \sum_{lk} B^l_{i,j}(r)B^k_{i,j}(r)$$
$$+ \sum_{lkp} B^l_{i,j}(r)B^k_{i,j}(r)B^p_{i,j}(r) + (-1)^{j-1}B^i_{i,j} \cdot \dots \cdot B^{j-1}_{i,j}$$

Proof. For the offered traffic of traffic type r with i as a source node and j as a destination node, $A_{i,j}(r)$, amount of non-blocked traffic at the destination node is equal to $(1 - B^i_{i,j}(r))(1 - B^{i+1}_{i,j}(r)) \cdot \dots \cdot (1 - B^{j-1}_{i,j}(r)) \cdot A_{i,j}(r)$. Therefore, the amount of blocked traffic is $A_{i,j}(r) - (1 - B^i_{i,j}(r))(1 - B^{i+1}_{i,j}(r)) \cdot \dots \cdot (1 - B^{j-1}_{i,j}(r)) \cdot A_{i,j}(r)$, so that the total blocking probability $B_{i,j}(r)$ is

$$B_{i,j}(r) = 1 - (1 - B^i_{i,j}(r))(1 - B^{i+1}_{i,j}(r)) \cdot \dots \cdot (1 - B^{j-1}_{i,j}(r)) =$$
$$= \sum_l B^l_{i,j}(r) - \sum_{lk} B^l_{i,j}(r)B^k_{i,j}(r)$$
$$+ \sum_{lkp} B^l_{i,j}(r)B^k_{i,j}(r)B^p_{i,j}(r) + (-1)^{j-1}B^i_{i,j} \cdot \dots \cdot B^{j-1}_{i,j}$$

□

We first find $B^l_{i,j}(r)$ along the path starting with the hop 0. Given $A_{0,j}(r)$, which is equal to $A^0_{0,j}(r)$, for all possible destinations $1 \le j \le k+1$ we find the total offered traffic of type r on link 0 as $A^0(r) = \sum_j A_{0,j}(r)$. We then find $B^0(r)$ using the algorithms presented in a previous section. For any destination $1 \le j \le k+1$, $B^0_{0,j}(r)$ is then equal to

$$B_{0,j}^{0}(r) = \frac{A_{0,j}^{0}(r)}{A^{0}(r)} \cdot B^{0}(r)$$

For hop 1, $A_{0,j}^{1}(r) = (1 - B_{0,j}^{0}(r))A_{0,j}^{0}(r)$, $A_{1,j}^{1}(r) = A_{1,j}(r)$ and $A^{1}(r) = \sum A_{0,j}^{1}(r) + \sum A_{1,j}^{1}(r)$ and we apply the algorithms of previous section to find $B^{1}(r)$. Then

$$B_{i,j}^{1}(r) = \frac{A_{i,j}^{1}(r)}{A^{1}(r)} \cdot B^{1}(r)$$

for $i = 0, 1$ and $2 \leq j \leq k + 1$.

In general, for hop l, $l \geq 1$, $A_{i,j}^{l}(r) = (1 - B_{i,j}^{l-1}(r))A_{i,j}^{l-1}(r)$ for $i < l$ or $A_{l,j}^{l}(r) = A_{l,j}(r)$ for $i = l$, $A^{l}(r) = \sum A_{i,j}^{l}(r)$.

$$B_{i,j}^{l}(r) = \frac{A_{i,j}^{l}(r)}{A^{l}(r)} \cdot B^{l}(r)$$

Once we obtain all the $B_{i,j}^{l}(r)$, $i \leq l < j$, we can find the total $B_{i,j}(r)$ as

$$B_{i,j}(r) = \bigcup_{l=i}^{j-1} B_{i,j}^{l}(r) = \sum_{l} B_{i,j}^{l}(r) - \sum_{lk} B_{i,j}^{l}(r) B_{i,j}^{k}(r)$$

$$+ \sum_{lkp} B_{i,j}^{l}(r) B_{i,j}^{k}(r) B_{i,j}^{p}(r) + (-1)^{j-1} B_{i,j}^{i} \cdot \dots \cdot B_{i,j}^{j-1}$$

For the 2-hop path represented on Figure 6.2, if offered traffic for every traffic type is known, we can find the blocking probabilities for a traffic type r.

Total offered traffic on link 0 is equal to $A^{0}(r) = A_{01}^{0}(r) + A_{02}^{0}(r)$, with $A_{01}^{0}(r) = A_{01}(r)$ and $A_{02}^{0}(r) = A_{02}(r)$ from which we can find the blocking probability on link 0, $B^{0}(r)$. Then

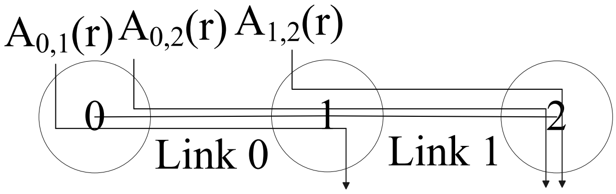

Figure 6.2: A 2-hop path.

$$B^0_{01}(r) = \frac{A^0_{01}(r)}{A^0(r)} \cdot B^0(r)$$

$$B^0_{02}(r) = \frac{A^0_{02}(r)}{A^0(r)} \cdot B^0(r)$$

For link 1, $A^1(r) = A^1_{02}(r) + A^1_{12}(r)$ where $A^1_{02}(r) = (1 - B^0_{02}(r))A^0_{02}(r)$ and $A^1_{12}(r) = A_{12}(r)$.

When we find blocking probability on link 1, $B^1(r)$, then

$$B^1_{02}(r) = \frac{A^1_{02}(r)}{A^1(r)} \cdot B^1(r)$$

$$B^1_{12} = \frac{A^1_{12}(r)}{A^1(r)} \cdot B^1(r)$$

The total probabilities for all source destination pairs are

$$B_{01}(r) = B^0_{01}(r)$$

$$B_{02}(r) = B^0_{02}(r) + B^1_{02}(r) - B^0_{02}(r) \cdot B^1_{02}(r)$$

$$B_{12}(r) = B^1_{12}(r)$$

For the 3-hop path represented on Figure 6.3 knowing all offered traffic for every traffic type we can find blocking probabilities for traffic type r as follows.

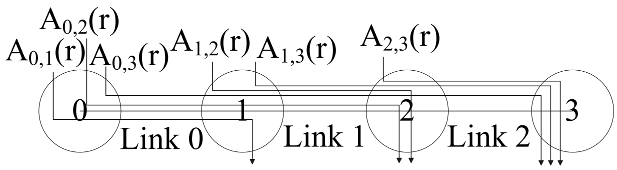

Figure 6.3: A 3-hop path.

The total offered traffic on link 0 is:

$$A^0(r) = A^0_{01}(r) + A^0_{02}(r) + A^0_{03}(r)$$

with $A^0_{01}(r) = A_{01}(r)$ and $A^0_{02}(r) = A_{02}(r)$ $A^0_{03} = A_{03}(r)$ from which we can find the blocking probability on link 0, $B^0(r)$. Then,

$$B^0_{01}(r) = \frac{A^0_{01}(r)}{A^0(r)} \cdot B^0(r)$$

$$B^0_{02}(r) = \frac{A^0_{02}(r)}{A^0(r)} \cdot B^0(r)$$

$$B^0_{03}(r) = \frac{A^0_{03}(r)}{A^0(r)} \cdot B^0(r)$$

For link 1

$$A^1(r) = A^1_{02}(r) + A^1_{03}(r) + A^1_{12}(r) + A^1_{13}(r)$$

where $A^1_{02}(r) = (1-B^0_{02}(r))A^0_{02}(r)$, $A^1_{03} = (1-B^0_{03}(r))A^0_{03}(r)$, $A^1_{12}(r) = A_{12}(r)$ and $A^1_{13}(r) = A_{13}(r)$. When we find blocking probability on link 1 $B^1(r)$, then

$$B^1_{02}(r) = \frac{A^1_{02}(r)}{A^1(r)} \cdot B^1(r)$$

$$B^1_{03}(r) = \frac{A^1_{03}(r)}{A^1(r)} \cdot B^1(r)$$

$$B^1_{12}(r) = \frac{A^1_{12}(r)}{A^1(r)} \cdot B^1(r)$$

$$B^1_{13}(r) = \frac{A^1_{13}(r)}{A^1(r)} \cdot B^1(r)$$

For link 2

$$A^2(r) = A^2_{03}(r) + A^2_{13}(r) + A^2_{23}(r)$$

where $A^2_{03}(r) = (1 - B^1_{03}(r))A^1_{03}(r)$, $A^2_{13}(r) = (1 - B^1_{13}(r))A^1_{13}(r)$ and $A^2_{23}(r) = A_{23}(r)$. When we find blocking probability on link 2, $B^2(r)$

$$B^2_{03}(r) = \frac{A^2_{03}(r)}{A^2(r)} \cdot B^2(r)$$

$$B^2_{13}(r) = \frac{A^2_{13}(r)}{A^2(r)} \cdot B^2(r)$$

$$B^2_{23}(r) = \frac{A^2_{23}(r)}{A^2(r)} \cdot B^2(r)$$

Finally, the total probabilities for all source destination pairs are

$$B_{01}(r) = B_{01}^{0}(r)$$

$$B_{02}(r) = B_{02}^{0}(r) + B_{02}^{1}(r) - B_{02}^{0}(r) \cdot B_{02}^{1}(r)$$

$$B_{03}(r) = B_{03}^{0}(r) + B_{03}^{1}(r) + B_{03}^{2}(r) - B_{03}^{0}(r) \cdot B_{03}^{1}(r) -$$

$$-B_{03}^{0}(r) \cdot B_{03}^{2}(r) - B_{03}^{1}(r) \cdot B_{03}^{2}(r) + B_{03}^{0}(r) \cdot B_{03}^{1}(r) \cdot B_{03}^{2}(r)$$

$$B_{12}(r) = B_{12}^{1}(r)$$

$$B_{13}(r) = B_{13}^{1}(r) + B_{13}^{2}(r) - B_{13}^{1}(r) \cdot B_{13}^{2}(r)$$

$$B_{23}(r) = B_{23}^{2}(r)$$

6.3 Chapter Summary

In this chapter we have considered calculating blocking probabilities for several bandwidth allocation policies in a multi rate multi class networks. A request can require multiple units of each resource (the multi-rate case) and it does not have to be the same for different requests of the same traffic type. We have considered the standard complete-sharing policy, guaranteed-minimum sharing policies and a preemptive policy in a prioritized multi class system. Our goal was to determine the probability of a new customer being admitted with a desired grade of service. We have presented approximate analytical tools to obtain blocking probabilities in a multi rate multi class system, where users of the same class can have different resource requirements [55, 56]. We have also expanded our single link model to calculate blocking probabilities for a multihop path, when the offered traffic of each source destination pair along the path is known.

Chapter 7

Validation via Case Studies

7.1 Comparative Analysis of Preemptive Algorithms

7.1.1 The Single Link Model

Simulation Model and Experiments

We have written a simulation program in C to study preemption and to compare our two proposed fast algorithms to the two existing optimal connection preemption algorithms. In this study we are interested to see the behavior of proposed algorithms in case of a fully utilized link where an incoming request requires preemption in order to be satisfied. The major issue addressed in this simulation study is the applicability of choosing connections to be preempted randomly compared to choosing them optimally with respect to optimization objectives: excess bandwidth preempted, number of preempted connections and priorities of preempted connections.

We use a capacity corresponding to OC-1, without loss of generality. We generate connections until the link is full. Since it is hard to get traffic mix from real MPLS networks we chose to generate priority and bandwidth of connections randomly. For this experiment we chose four priority levels and tried to have a large number of calls for each priority. The distribution of the priority levels is uniform. The bandwidth range for connections is taken to be between 64 Kbps and 4000 Kbps as in the paper by Peyravian and Kshemkalyani [43], where the maximum value of the required bandwidth corresponds to 8-10% of the OC-1 link capacity . We used two distributions of bandwidth within this range, namely, Uniform and Pareto distributions, since network activity tends to have a heavy-tailed distribution and distribution looks like Pareto with $0.9 < \beta < 1.0$ [21]. The performance of the algorithms is affected by the distribution of the traffic mix. This choice of distributions can be seen as the best and worst case with respect to excess bandwidth preempted or number of preempted connections.

When a connection is generated, the program checks whether there is sufficient bandwidth available on the link to satisfy the bandwidth requested. If enough free bandwidth is available, the requested bandwidth is allocated and the request is accepted on the link. If not enough bandwidth is available, the program then uses a preemption selection algorithm (i.e., Min_BW, Min_Conn, RandPreempt or MinRandPreempt as described earlier) to select calls for preemption.

Our program collects and reports a number of performance measures that indicate how well a call preemption selection algorithm performs. These performance measures are as follows:

- Excess Bandwidth Preemption: This measure indicates how much excess bandwidth a preempting call of priority i preempts per each preemption.

- Link Utilization After Preemption.
- Number of Preemptions: This is the number of calls that a preempting call of priority i preempts in order to be setup on a link, per each preemption.
- Connection Preemption Probability: This is the probability that a connection of a given priority is preempted by a higher priority connection. It is important that the connection preemption probability decreases as the connection's priority increases, as we want the high-priority connections to be disturbed less.

To study the performance of the connection preemption algorithms discussed previously, we conducted 24 independent runs of the simulation program for each algorithm. In each run we executed a connection preemption algorithm for 10,000 times on a fully utilized link. The simulation program requires a simulation seed and the independence of the runs was achieved through the linear congruential model to generate simulation seeds.

We used confidence intervals to quantify the accuracy of performance measures obtained from the simulation experiments. Since the runs are independent, we can assume identical distribution of the replications. As the number of replications is small, we can assume that the mean is distributed as student-t distribution and calculate the $100(1-\alpha)\%$ confidence interval on the mean from the replications using [38]

$$\bar{x} - t_{\alpha/2}\frac{s}{\sqrt[2]{r}} < \mu < \bar{x} + t_{\alpha/2}\frac{s}{\sqrt[2]{r}} \tag{7.1}$$

where $\bar{x}$ is the sample mean, μ is the true mean, s is the sample standard deviation, R is the number of replications, and $t_{\alpha/2}$ is the critical value of the student-t distribution with

$(r-1)$ degrees of freedom. For this experiment we obtained the 95% confidence interval for all the performance measures. The $t_{\alpha/2}$ value for the 95% confidence interval when $r = 24$ is 2.069 which is obtained from [58].

Results and Discussion

This section presents the results that we obtained through simulation by comparing algorithms RandPreempt and MinRandPreempt to optimal algorithms Min_BW and Min_Conn.

Tables 7.1 and 7.2 show Excess Bandwidth Preemption, Average link reservation level after preemption and Number of Preemptions, for uniform and Pareto distribution of the traffic, respectively. For the Pareto distribution of traffic on the link the effect of complexity of Min_BW became apparent. Because of the excessive number of connections on the link available for preemption, and the exhaustive nature of Min_BW, the number of computations increased in an explosive manner, and our simulation did not finish in this case within the pre-specified runtime budget. For Pareto distribution we present results for both Min_Conn, RandPreempt and MinRandPreempt.

Looking at the confidence intervals for the performance measures we observe that, overall, 95% confidence intervals are small. Thus, the average values provide a good measure for comparison purposes.

Figures 7.1 and 7.2 show the probability density function of excess bandwidth preemption and number of preemptions in case of uniform distribution of traffic. The density functions of excess bandwidth preemption show that algorithms with random preemption preempt more bandwidth than needed comparing to optimal algorithms Min_BW and

	Min_BW	Min_Conn	Rand Preempt	Min Rand Preempt
Excess Bandw. Preempt. (Kbps)	220± 0.60	354± 1.04	1540± 4.51	1170± 3.41
Link Util. After Preempt. (%)	99.56± 0.008	99.30± 0.008	97.03± 0.008	97.75± 0.008
Number of Preempt.	1.68± 0.003	1.03± 0.003	1.41± 0.003	1.69± 0.003

Table 7.1: Performance Measures for Uniform Distribution of Traffic.

Min_Conn for uniform distribution of traffic. Density function of number of preemptions of all four algorithms is very similar.

Figures 7.3 and 7.4 show the probability density function of excess bandwidth preemption and number of preemptions in case of Pareto distribution of traffic. The density function of excess bandwidth preemption in case of RandPreempt and MinRandPreempt algorithms is very close to the density function of the Min_Conn algorithm. The density functions of the number of preemptions show that our algorithms with random preemption tend to preempt more connections compared to optimal algorithm Min_Conn.

Tables 7.3 and 7.4 show the connection preemption probability per priority level for each algorithm in case of uniform and Pareto distribution of the traffic, respectively.

Excess Bandwidth Preemption: By looking at the numbers in row 1 of Table I and Table II, we see that Min_BW preempts the least excess bandwidth, even less than

	Min_Conn	Rand Preempt	Min Rand Preempt
Excess Bandw. Preempt. (Kbps)	190± 1.56	444± 3.67	131± 1.09
Link Util. After Preempt. (%)	99.63± 0.007	99.15± 0.007	99.75± 0.007
Number of Preempt.	1.04± 0.001	3.29± 0.002	5.32± 0.003

Table 7.2: Performance Measures for Pareto Distribution of Traffic.

Min_Conn, as expected from the work of Peyravian and Kshemkalyani. RandPreempt preempts more excess bandwidth than MinRandPreempt which was a reason to select two connections at once and then choose to preempt the one occupying smaller amount of bandwidth. This measure itself cannot indicate how well an algorithm performs. As it can be seen from Tables I and II, excess bandwidth preemption very much depends on the traffic distribution in the network. For uniform distribution of traffic, MinRandPreempt preempts more excess bandwidth than Min_Conn, but in the case of Pareto distribution it actually preempts less excess bandwidth than Min_Conn.

Average Link Reservation Level After Preemption: To have more information on the effect of preemption on the available bandwidth of the link when using preemption we look at the average link reservation level after preemption is performed. As expected, Min_BW keeps the link almost fully utilized after preemption is performed, but RandPre-

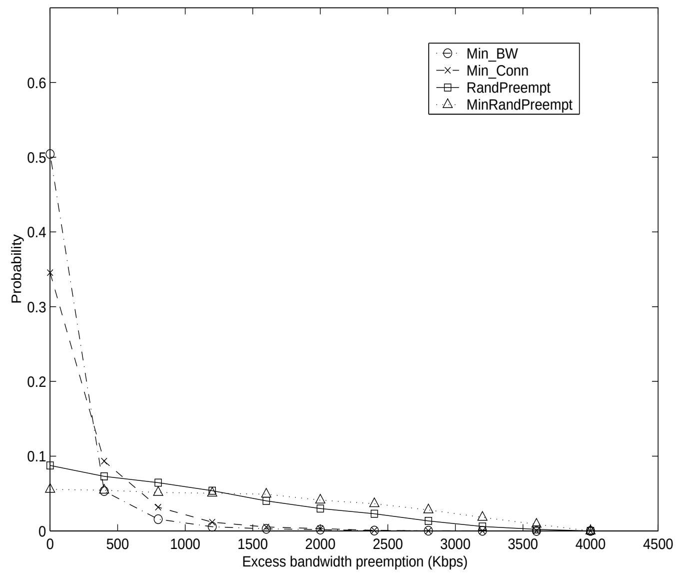

Figure 7.1: Probability density function of excess bandwidth preemption for uniform distribution of traffic.

empt and MinRandPreempt are also keeping the link highly utilized though they do not aim at the optimality when selecting the connections which will be preempted. In case of Pareto distribution of traffic, the link is almost fully utilized after preemption. MinRandPreempt keeps the utilization of the link higher than RandPreempt by selecting two connections at once and then preempting the one occupying the smaller amount of bandwidth.

Number of Preemptions: For all algorithms in case of uniform distribution is small. Min_Conn as expected performs the best, but RandPreempt and MinRandPreempt are not far behind. In case of Pareto distribution of the traffic Min_Conn has the best performance but now RandPreempt and MinRandPreempt preempt more connections because of the increase in number of connections with lower priorities available for preemption.

Connection Preemption Probability: From Tables III and IV we see that, for all algorithms, the connection preemption probability decreases as the connection's priority

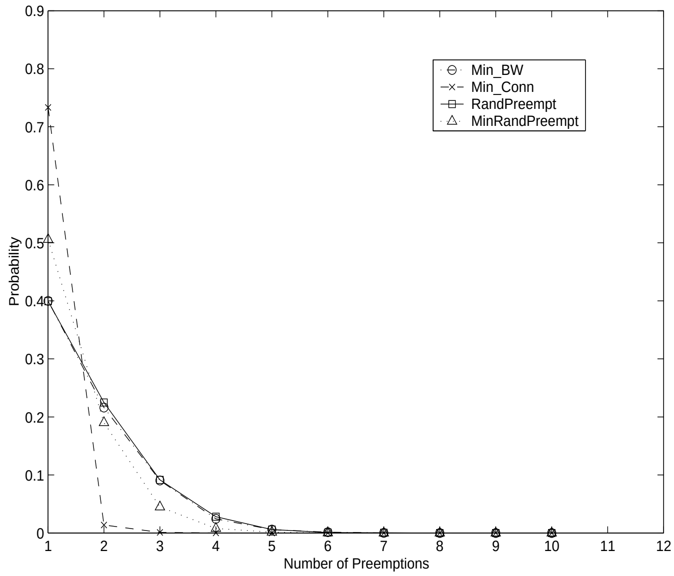

Figure 7.2: Probability density function of number of preemptions for uniform distribution of traffic.

increases even with random selection of connections for preemptions in RandPreempt and MinRandPreempt. This is very desirable as we want the high-priority connections to be disturbed less. There is no significant difference among all connection preemption selection algorithms.

7.1.2 A Dynamic Environment Study

Simulation Model and Experiments

In this section, simulation results are shown to compare connection preemption algorithms in a dynamic environment. In [43] the authors conclude that, in terms of overall network performance, there is no significant difference between Min_BW and Min_Conn. Furthermore, as we observed in [52], algorithm Min_BW can be so computationally complex and time consuming, as to exceed any reasonable time budget, under even relatively simple

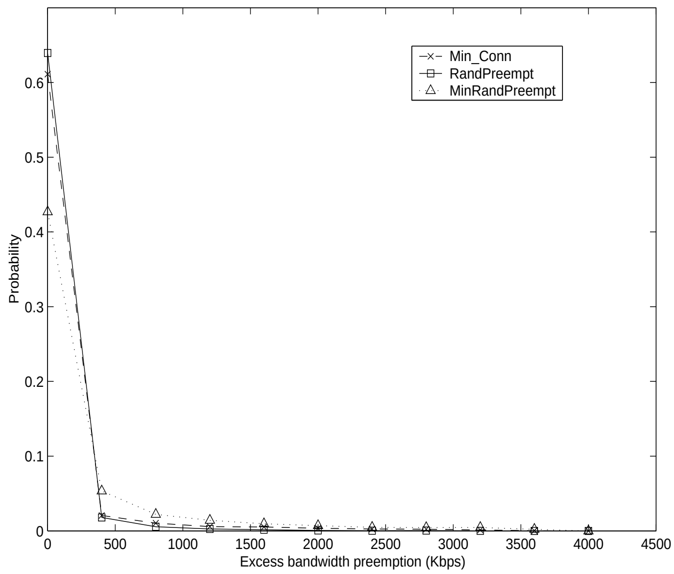

Figure 7.3: Probability density function of excess bandwidth preemption for Pareto distribution of traffic.

scenarios. For these reasons we compare our algorithms to Min_Conn only, in order to finish the simulation within the pre-specified runtime budget.

We have written a simulation program in C to study preemption and to compare our two proposed fast algorithms to the existing optimal connection preemption algorithm. The program simulates the lives of the calls from the time they are created until the time they terminate. A new call is generated according to an exponential arrival rate. A call generation can have several outcomes: the setup can be successful or unsuccessful or the call can be preempted by another higher priority call and rerouted. If a call is rerouted, the reroute can be successful, or the call can be dropped. In the simulations, the network topology shown in Figure 7.5 similar to [57] was used. The primary path for all calls is OC-1 and a path to which a call is rerouted if preempted, or if it cannot be established on the primary path, is 15Mbps.

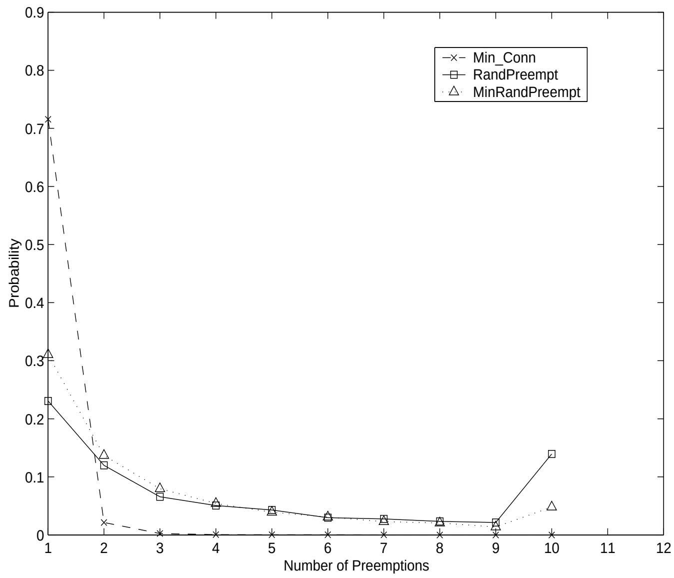

Figure 7.4: Probability density function of number of preemptions for Pareto distribution of traffic.

The priority and bandwidth of connections using the link are chosen probabilistically. For this experiment we chose four priority levels, with 0 being the highest priority and 3 the lowest, and tried to have a large number of calls for each priority. The distribution of the priority levels is uniform. The bandwidth range for connections is taken to be between 64 Kbps and 4000 Kbps. We used uniform and the Pareto distribution for bandwidth requirements within this range, since network activity tends to have a heavy-tailed distribution and distribution resembles Pareto with $0.9 < \beta < 1.0$ [21]. This choice of distributions can be seen as corresponding the best and worst case with respect to excess bandwidth preempted or number of preempted connections. The connection holding times are assumed to be exponentially distributed with a mean time of 600 seconds. The connection arrival rate to the network has an exponential distribution with a mean of about 1.5 connections/second as in [43].

Call Priority	Min_BW	Min_Conn	Rand Preempt	Min Rand Preempt
0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
1	$0.132\pm$ 0.001	$0.109\pm$ 0.001	$0.111\pm$ 0.001	$0.114\pm$ 0.001
2	$0.293\pm$ 0.001	$0.275\pm$ 0.001	$0.281\pm$ 0.001	$0.283\pm$ 0.001
3	$0.575\pm$ 0.001	$0.616\pm$ 0.001	$0.608\pm$ 0.001	$0.603\pm$ 0.001

Table 7.3: Connection Preemption Probability for Uniform Distribution of Traffic.

Call Priority	Min_Conn	Rand Preempt	Min Rand Preempt
0	0 ± 0	0 ± 0	0 ± 0
1	$0.109\pm$ 0.001	$0.110\pm$ 0.001	$0.118\pm$ 0.001
2	$0.268\pm$ 0.001	$0.278\pm$ 0.001	$0.281\pm$ 0.001
3	$0.623\pm$ 0.001	$0.612\pm$ 0.001	$0.601\pm$ 0.001

Table 7.4: Connection Preemption Probability for Pareto Distribution of Traffic.

When a connection is generated, the program checks whether there is sufficient bandwidth available on the primary path to satisfy the bandwidth requested. If enough free bandwidth is available, the requested bandwidth is allocated and the request is accepted on the OC-1 link. If not enough bandwidth is available, the program then uses a preemption selection algorithm (i.e., Min_Conn, RandPreempt or MinRandPreempt) to select calls for preemption and reroutes the preempted calls on the secondary path.

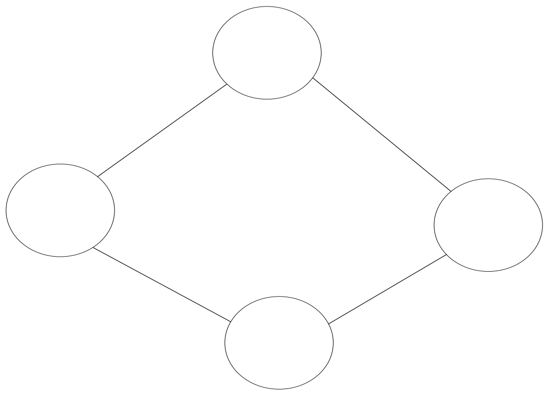

Figure 7.5: Baseline network topology for dynamic environment study.

Our program collects and reports a number of performance measures that indicates how well a call preemption selection algorithm performs. These performance measures are as follows:

- Excess Bandwidth Preemption: This measure indicates how much excess bandwidth a call of priority i preempts per each instance of preemption.
- Average Link Reservation Level.
- Number of Preemptions: This is the number of calls that a call of priority i preempts in order to be setup on a link per each preemption.
- Completion Probability: This is the probability that a call of priority i is completed.

To study the dynamic performance of the connection preemption algorithms discussed previously, we conducted 24 independent runs of the simulation program for each algorithm. In each run we executed a connection preemption algorithm for 10,000 seconds. The independence of the runs was achieved through the linear congruential model to generate simulation seed for the simulation.

In the study we have used confidence interval technique to quantify the accuracy of performance measures obtained from the simulation experiments and the stability of all algorithms. Since the runs are independent, we can assume identical distribution of the replications. As the number of replications is small, we can assume that the mean is distributed as student-t distribution and calculate the $100(1-\alpha)\%$ confidence interval on the mean from the replications [38]. We obtained the 95% confidence interval for all the performance measures.

Results and Discussion

This section presents the results that we obtained through simulation by comparing the dynamic performance of algorithms RandPreempt and MinRandPreempt to that of optimal algorithm Min_Conn. Tables 7.5 and 7.6 show Excess Bandwidth Preemption, Averaged link reservation level and Number of Preemptions for each algorithm in case of uniform and Pareto distribution of the traffic, respectively.

Tables 7.7 and 7.8 show the completion probability per priority level for each algorithm in case of uniform and Pareto distribution of the traffic, respectively.

Excess Bandwidth Preemption: Regarding the excess bandwidth preemption, we see that RandPreempt preempts more excess bandwidth than MinRandPreempt which was a reason to select two connections at once and then choose to preempt the one occupying smaller amount of bandwidth.

Average Link Reservation Level: For the average link reservation level we observe that RandPreempt and MinRandPreempt are keeping the links highly utilized though they

	Min_Conn	Rand Preempt	Min Rand Preempt
Excess Bandw. Preempt. (Kbps)	350.612± 5.54	480.665± 5.77	468.81± 4.49
Avg. Link resv. level (%)	99.69± 0.001	98.81± 0.001	98.84± 0.001
Number of Preempt.	1.53± 0.01	1.62± 0.01	1.61± 0.01

Table 7.5: Performance Measures for Uniform Distribution of Traffic.

do not aim at the optimality when selecting the connections which will be preempted.

Number of Preemptions: The number of preemptions for all algorithms is small. Min_Conn, as expected, performs the best, but RandPreempt and MinRandPreempt are not far behind.

Completion Probability: Finally, with respect to the completion probability, from Tables 7.7 and 7.8 we see that for all algorithms the completion probability increases as the connection's priority increases, as expected from the use of preemption.

We can see that in the case of a dynamic setting, there is practically no difference in performance between the optimal algorithm and the algorithms with random selection of calls for preemption. In the previous study we observed that the probability that a connection of a given priority is preempted by a higher priority connection decreases as the connection's priority increases for all algorithms. Over time, the set of connections using

	Min_Conn	Rand Preempt	Min Rand Preempt
Excess Bandw. Preempt. (Kbps)	58.83± 0.93	122.01± 1.77	115.85± 1.08
Avg. Link resv. level (%)	98.58± 0.001	98.24± 0.001	98.26± 0.001
Number of Preempt.	1.56± 0.01	1.79± 0.01	1.80± 0.01

Table 7.6: Performance Measures for Pareto Distribution of Traffic.

the link with lower priority than the preempting connection is exhausted by the arrival of higher priority connections. When the high priority request rate is intense, i.e., connection interarrival times are small compared to connection holding times, the algorithms produce similar results. The completion probability for higher priorities is very close to 1, while for lower priorities is very low, because there are only two paths from source to destination and connection preemption provides higher network availability for calls of higher priority than others.

7.1.3 Full Network Environment Study

Simulation Model and Experiments

In this experiment we used a connection-level simulation to study preemption and to compare connection preemption algorithms in a full network environment. The simulation

Call Priority	Min_Conn	Rand Preempt	Min Rand Preempt
0	1 ± 0	1 ± 0	1 ± 0
1	0.67± 0.09	0.62± 0.008	0.65± 0.007
2	0.1± 0.004	0.06± 0.003	0.05± 0.003
3	0.05± 0.002	0.05± 0.002	0.02± 0.002

Table 7.7: Completion Probability per Priority Level for Uniform Distribution of Traffic.

Call Priority	Min_Conn	Rand Preempt	Min Rand Preempt
0	1 ± 0	1 ± 0	1 ± 0
1	0.64± 0.012	0.66± 0.008	0.71± 0.007
2	0.08± 0.004	0.06± 0.003	0.05± 0.003
3	0.04± 0.002	0.03± 0.002	0.02± 0.002

Table 7.8: Completion Probability per Priority Level for Pareto Distribution of Traffic.

model has most mechanisms of typical connection-oriented networks. The input to the simulation program includes a network configuration: the nodes, the transmission links with their capacities, source/destination distribution, connections characteristics, and other controlling parameters such as simulation time, simulation seeds, and maximum connection hops. The program collects and reports a number of statistics as will be described later.

In this experiment also, we simulated the lives of connections from the time they are created until they terminate. Connection interarrival times are exponentially distributed. Upon arrival of a connection to the network, its source and destination nodes, traffic type, bandwidth, and holding time are chosen probabilistically. Once the connection parameters are selected, a path selection algorithm is run and a path in the network is determined. This algorithm attempts to find *the minimum weight* path. Each link in the network is associated with an increasing function of the load of the link to reconcile the contradiction in the selection of the shortest path and load balancing as described in [27], [1]. The weight of a path is the sum of weights of the individual links on the path. A path with a low weight indicates a path with a low load in terms of this measure.

When a connection request arrives, a connection is established if the network has the bandwidth to support it. If there is not enough bandwidth to establish the connection, but if there are sufficient low-priority connections that can be preempted to free enough bandwidth for this connection, then those low-priority connections will be preempted and the connection request gets satisfied. If the connection request cannot be accommodated, it is rejected. When a connection is preempted, it is treated like a new connection. After a connection is successfully completed, it is taken down.

In the simulations, the two-tiered network topology shown in Figure 7.6 from [16] was used. We used a capacity corresponding to OC-1 for all links, without loss of generality.

The connection holding times are assumed to be exponentially distributed with a mean of 600 seconds. The source and destination for the connections were selected randomly with a uniform distribution such that the load in the network is uniformly distributed. For this experiment we implemented eight priority levels, with 0 being the highest priority and

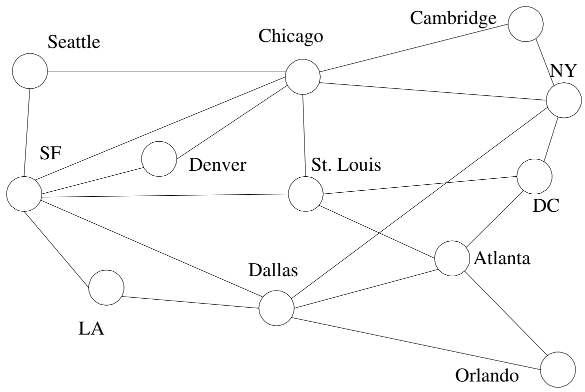

Figure 7.6: Baseline network topology for a full network environment study.

7 the lowest, and tried to have a large number of calls for each priority. The distribution of the priority levels is uniform.

In our experiments we were interested in how different performance metrics change as the network load increases. Therefore, we were loading the network to the level of causing connection preemption events, in order for new, higher priority connections to be established. Performance measures are as follows:

- Average link reservation level of the network
- Number of unsatisfied requests: This is the number of unsatisfied requests either because of lack of resources to establish the connection or unsuccessfully rerouted preempted connections. The lower the number of unsatisfied requests, the higher the connection success probability was.
- Number of Preemptions: This is the number of calls that a preempting call of priority i preempts in order to be setup on a link per each preemption.

- Probability of preemption: This is the probability that a connection of a given priority is preempted by a higher priority connection.

Results and Discussion

In this section, simulation results are shown to compare connection preemption algorithms in a network environment.

Network Performance as a function of arrival rate: Figure 7.7 shows the *number of unsatisfied requests* as a function of the connection arrival rate with and without preemption. The figure also shows the analysis of the effect of preemption in terms of network throughput. The top curve is for the non-preemption case, the two middle curves are for the random selection algorithms, and the bottom curve is for the *Min_Conn* algorithm.

In all the cases, we see that when the connection arrival rate is low, i.e., when the network is lightly loaded, the number of unsatisfied requests is almost zero, as expected. As the connection arrival rate increases, the number of unsatisfied requests increases because the network now has less free bandwidth. When the connection arrival rate is low, the effect of preemption is very small, i.e., there is no significant difference between the preemption and the non-preemption cases in terms of the number of unsatisfied requests. However, as the load in the network increases due to more connection requests, the advantage of preemption becomes more obvious, as more connection requests are satisfied.

Connection preemption, coupled with the capability to reroute preempted connections, offers resources in a more dynamic way. It improves the path selection and allows correction of a "wrong" decision made previously to accept a "less valuable" connection. As far as the performance of the connection preemption selection algorithms is concerned, there is no significant difference. In essence, the Min_Conn algorithm performs only slightly better than the random selection algorithms and the performance difference slightly increases with the connection arrival rate. The reason for this is that Min_Conn algorithm causes a smaller number of preemptions than the random selection algorithms, thus "inducing" less rerouting in the network as shown on figure 7.8.

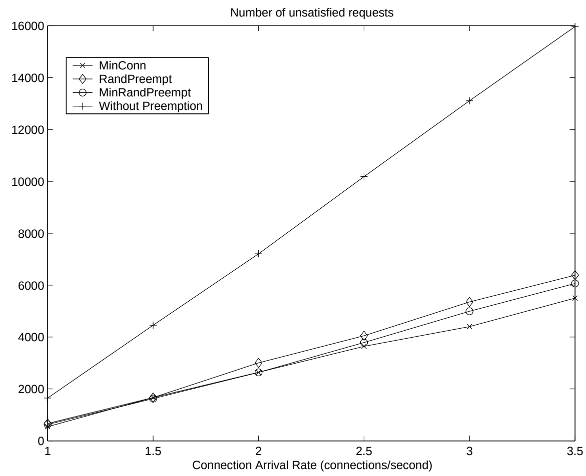

Figure 7.7: Number of unsatisfied requests as a function of arrival rate.

Network behavior as a function of time: Fig. 7.9 shows the average link reservation level versus the simulation time for one of the experiments. The connection arrival rate to the network for this experiment has an exponential distribution with a mean of about 2.5 connections/second. Initially, there is no connection in the network, so the average link reservation level is equal to zero. As connections arrive, the average link reservation level rises. After a short time of about 1500 seconds, the average link reservation level reaches a stable level, approximately 80% for the average link reservation level. Using bandwidth reallocation and rerouting of the preempted connections we increase the network utilization,

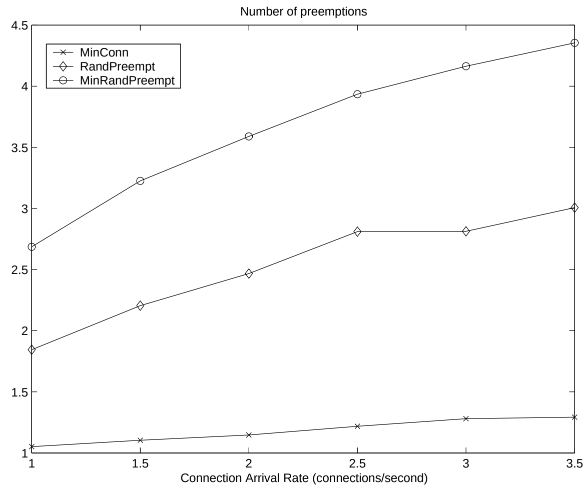

Figure 7.8: Number of preemptions as a function of arrival rate.

by satisfying more requests, and thus we increase the generated revenue of the network provider.

Performance metrics for priority levels: Fig. 7.10 shows the connection preemption probability for each priority level for one of the experiments. The figure is for the case when the connection arrival rate is approximately 1 connections/second. We see that for all the algorithms the connection preemption probability decreases as the connections priority increases causing the high-priority connections to be disturbed less. The *higher network availability* for connections of *higher priority* is the desirable and expected result from the use of preemption.

In these case studies we have compared our sub-optimal connection preemption algorithms that use random selection, to an existing optimal algorithm in three settings: single link, simple dynamic model and full network environment. The comparison between the optimal algorithms and our two connection preemption algorithms shows that, in terms

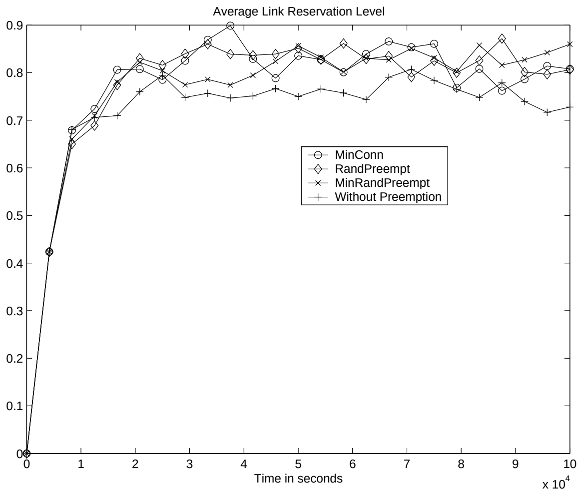

Figure 7.9: Network behavior in time.

of overall network performance, the difference is practically insignificant. The reason for this is that connection preemption algorithms operate in a decentralized network, i.e., they are executed independently in each link along the path. Over time the set of connections using the link with lower priority than the preempting connection is exhausted by the arrival of higher priority connections.

7.2 Dynamic Utility-based Bandwidth Allocation Policies

7.2.1 Overloaded Network

Simulation Model and Experiments

We have written a simulation program in C to study our dynamic utility based bandwidth allocation policies and compare them in a dynamic network environment where connections come and go. We are using the simulation model for the full network environ-

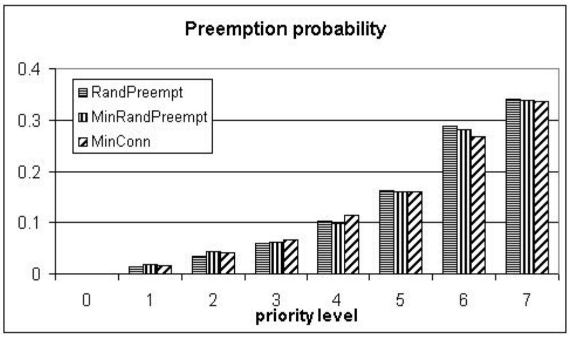

Figure 7.10: Connection Preemption Probability per Priority Level.

ment from the previous study.

In the simulations, the two-tiered network topology shown in Figure 7.6 was used. We used a capacity corresponding to OC-1 for all links, without loss of generality. Since it is hard to obtain the traffic mix from real MPLS networks we chose to generate bandwidth values of connections randomly. The bandwidth range for mean rate of connections is taken to be between 64 Kbps and 4,000 Kbps with uniform distribution as in the paper by Peyravian and Kshemkalyani [43], where the maximum value of the required bandwidth corresponds to 8-10% of the OC-1 link capacity. The effective bandwidth of connections is computed based on the model introduced in [27], [1] with other parameters such as active burst length assumed as 1 sec, utilization 0.9, buffer size is taken as R_k and Loss Ratio is 0.1% [51]. The network traffic is comprised of four service classes introduced in Section 5. The assumption on the distribution of the traffic mix can greatly affect the performance of the static guaranteed bandwidth allocation policies. As a general case we assume that application types are uniformly distributed among requests.

The connection holding times are assumed to be exponentially distributed with a mean of 600 seconds. The source and destination for the connections were selected randomly with a uniform distribution such that the load in the network is uniformly distributed.

In our experiments we were interested in how different performance metrics change as network load increases. Therefore, we were loading the network to the level of causing connection rerouting in order for new, higher priority connection to be established. Our program collects and reports a number of performance measures that indicates how well dynamic bandwidth allocation performs in respect to the objectives defined in Section 5.

Results and Discussion

In this section, simulation results are shown to compare dynamic bandwidth allocation policies in a network environment. A number of bandwidth allocation policies have been discussed in the literature. We chose to compare adaptive bandwidth allocation schemes to the most general case of *static* resource allocation, called a *complete sharing* (CS) admission policy, where all connections are admitted if resources are available at the time a connection is requested. As it is observed [43], algorithm Min_BW is so computationally complex and time consuming, as to exceed any reasonable time budget, under even relatively simple scenarios. For the heuristic we take into take into account the amount of bandwidth rerouted, the number of connections rerouted, and their priority equally when determining which connections to be rerouted.

Figure 7.11 shows the *number of unsatisfied requests* as a function of the connection arrival rate in case of the static and dynamic bandwidth allocation. As the load in the network increases due to more connection requests, the advantage of dynamic bandwidth allocation becomes obvious, as more connection requests are satisfied. Using dynamic bandwidth we can increase the network utilization, by satisfying more requests, and thus increase the generated revenue of the network provider.

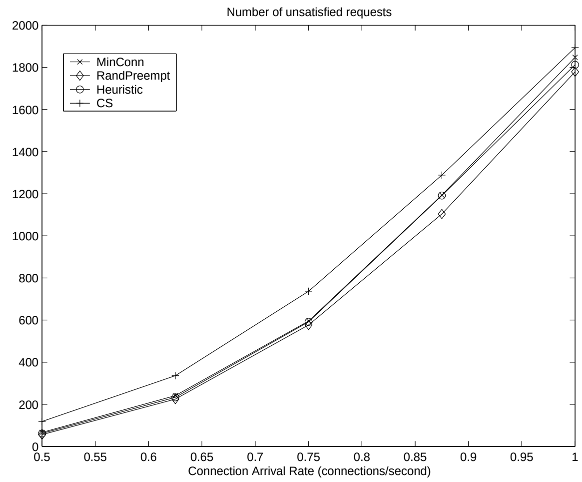

Figure 7.11: Number of unsatisfied requests as a function of arrival rate.

Figure 7.12 shows the distribution of unsatisfied requests for the case when connection arrival rate to the network for this experiment has a mean of about 1 connection/second. Dynamic bandwidth allocation allows a service provider to differentiate between different types of customers based on their priority or the service charges that they pay, in order to offer real time services, to provide QoS guarantees for multimedia traffic and to guarantee stability even in overloaded conditions.

Figure 7.13 shows the *total utility of satisfied requests* as a function of arrival rate. When the connection arrival rate is low there is no significant difference between the static and dynamic bandwidth allocation policies in terms of the total utility of satisfied requests. However, as the load in the network increases due to more connection requests, the advantage of dynamic bandwidth allocation becomes obvious from revenue generation

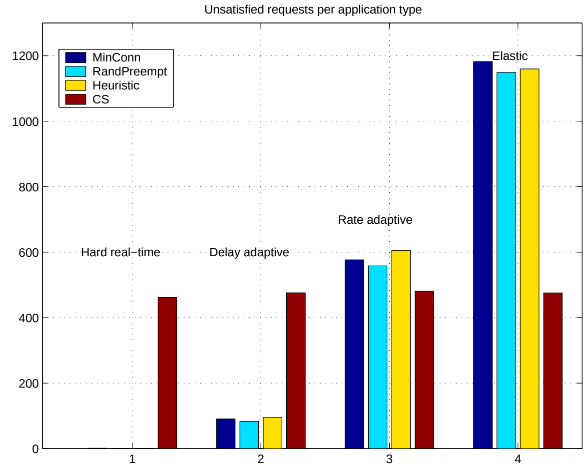

Figure 7.12: Distribution of unsatisfied requests per application type.

perspectives, as more connection requests are satisfied. As far as the performance of the dynamic algorithms is concerned, there is no significant difference.

Dynamic bandwidth allocation allows correction of a "wrong" decision made previously to accept a less valuable connection. As far as the performance of the dynamic algorithms is concerned, there is no significant difference. In essence, the Min_Conn and objective function heuristic algorithm perform only slightly better than the random selection algorithms and the performance difference slightly increases with the connection arrival rate. The reason for this is that these algorithms cause a smaller number of reroutes than the random selection algorithms, as shown on Figure 7.14. Random selection algorithms could be used when rerouting is allowed and not expensive, e.g. in simple topology networks.

Figure 7.15 shows the average link reservation level versus the simulation time for one of the experiments. The connection arrival rate to the network for this experiment has a mean of about 0.75 connections/second. Initially, there is no connection in the network,

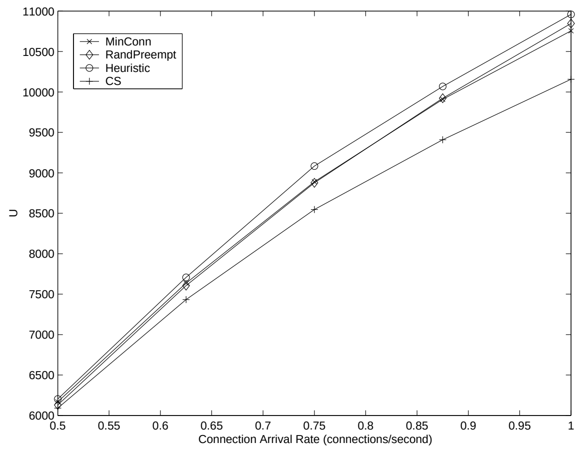

Figure 7.13: The total utility of satisfied requests as a function of the arrival rate.

so the average link reservation level is equal to zero. As connections arrive, the average link reservation level rises. After a short time of about 1,500 seconds, the average link reservation level reaches a stable level, approximately 75% for the average link reservation level. Dynamic bandwidth allocation improves the path selection, assuring that high priority requests are routed through relatively favorable paths within a differentiated services environment.

7.2.2 Topology variations

Simulation Model and Experiments

Network engineering needs to ensure that network connectivity is restored within a certain constrained amount of time, and performance levels so as not to affect the services transported. Any traffic-engineered network that carries critical, high-priority traffic needs to be resilient to faults of any type. Several scenarios can be explored through simulation.

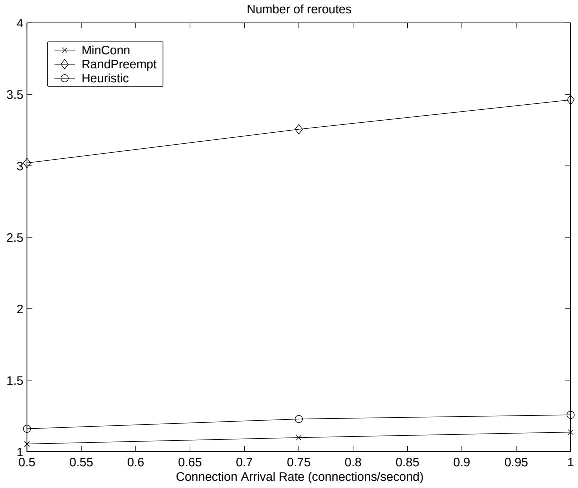

Figure 7.14: Number of reroutes as a function of arrival rate.

First, if an entire node is knocked out (e.g., complete system failure). This means that service to the node in question is impossible for any connection originating from that node, destined for that node or routed through the node in question. Clearly, the probability of such a complete loss is very low, particularly in large cities where providers maintain multiple points of presence. However, the recent tragic events in New York City and the blackout in the northeast states showed the impact of an important node sustaining damage. A second scenario might include the loss of a backbone provider which results in a drop of resources on links from that node. Finally, if select links in a network are severed, isolated nodes suffer loss of service, but nodes with multiple links remain functional [26].

For each node in the network we define a Capacity Degradation Ratio CDR_i. Let N_i denote the neighbors of node i. Let C_{ij} represent the finite capacity of link between nodes i and j.

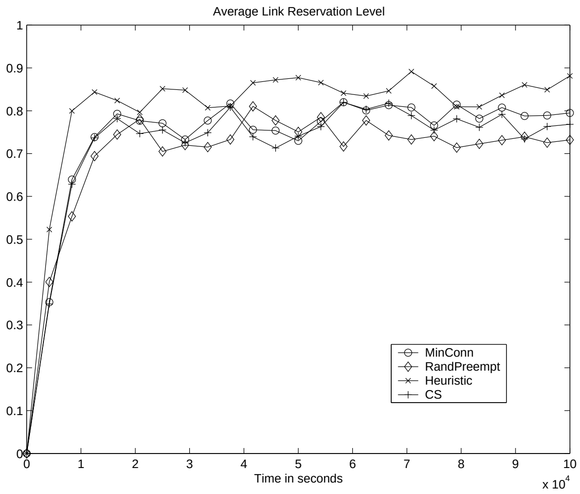

Figure 7.15: Network behavior in time.

$$CDR_i = \frac{\sum_{j \in N_i} C_{ij}}{\sum_{j \in N_i} C_{ij}(0)}$$

The weighted utility of the request is calculated at each node i as $u_k^{(i)} = CDR_i \cdot u_k$ representing the decrease in the utility per application type at node i as a result of capacity changes. In this study we focus on restoration of traffic in case of minor failures, which is typically caused by a single link failure, which occur most frequently up to major failures when multiple links fail or a single node fails. A number of connections with different source-destination node pairs which are affected due to the link failure depends on the number of failed links and their location in the network. In our experiments the location of the failed links is chosen based on the degree of the node. From Figure 7.6, we can see that the minimum node degree is 2 and the maximum is 6. We increase the number of the failed link from single link failure to complete node failure.

Results and Discussion

In this experiment we investigate the performance of our general dynamic bandwidth allocation scheme in case of link failures in the network. The links are made to fail around 2000 seconds into the simulation, once the network has reached a stable level. We assume that the load is uniformly distributed in the network. Obviously the impact of the failure(s) on the traffic in the network and thus the network performance will depend on the node degree where link or multiple links failed. We vary the number of links from one to complete node failure for three nodes with node degrees 2, 4 and 6. The network performance metrics are represented in Figures 7.16-7.19.

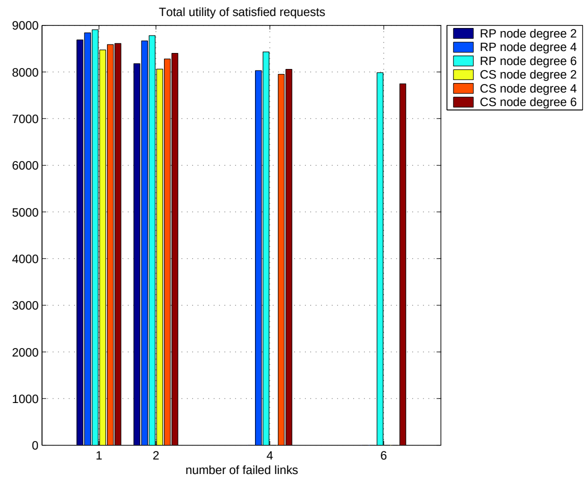

Figure 7.16: The total utility of satisfied requests as a function of number of failed links for different node degrees for dynamic (RP) and static (CS) bandwidth allocation.

Figure 7.16 shows the *total utility of satisfied requests* as a function of number of failed links for different node degrees, and for dynamic (RP) vs. static (CS) bandwidth allocation. When the number of failed links is low, the impact of the link failure is much larger on the *smaller degree* nodes, as expected with the total utility of satisfied request

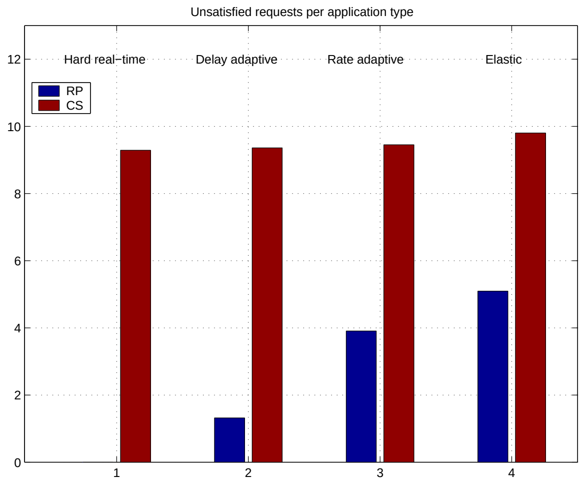

Figure 7.17: The number of unsatisfied requests per application type for the case of two failed links at node with degree 4, for dynamic (RP) and static (CS) bandwidth allocation.

decreasing as more links fail. Figures 7.17 and 7.19 show the distribution of unsatisfied requests and time delivery for connections affected by failure for the case of two failed links at a node with degree 4. Dynamic bandwidth allocation allows a service provider to differentiate between different types of customers based on their priority or the service charges that they pay, in order to offer real time services, to provide QoS guarantees for multimedia traffic and to guarantee QoS delivery even during failures.

7.2.3 Variable user requests

Simulation Model and Experiments

Uncoordinated user requests make up the third dimension of the dynamics in networks. This represents the classical teletraffic problem of a certain access point only having a limited available bandwidth which at times may be exceeded, if too many users are accessing it. If the bandwidth is large compared to the individual requirements, the

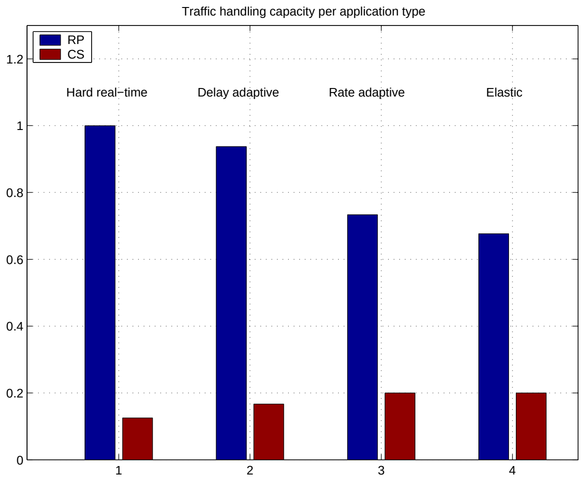

Figure 7.18: The traffic handling capacity of connections affected by failure per application type for the case of two failed links at node with degree 4, for dynamic (RP) and static (CS) bandwidth allocation.

total traffic load on each node is close to its average value over time and a very good resource utilization can be achieved. However, when user bandwidth requirements are high compared to the total available bandwidth and if the demands vary between users, a more flexible service model which allows dynamic resource allocation is needed.

No network can be economically designed for extreme overloads. Loads on public networks reach up to five times normal during an emergency causing important traffic to receive equally poor access to resources such as low priority traffic. In this scenario we investigate the distribution of the load throughout the topology, not the loading level of the network as a whole. In general, real networks are asymmetric both in traffic and capacity due to the asymmetric growth in traffic demand. Our model captures this concentration of the traffic in certain areas of the network which significantly impacts the performance of the network. If the load is uniformly distributed through the network with $|N|$ nodes, the average number of connections at link j with source s, $\bar{K}_j^s$ is equal to

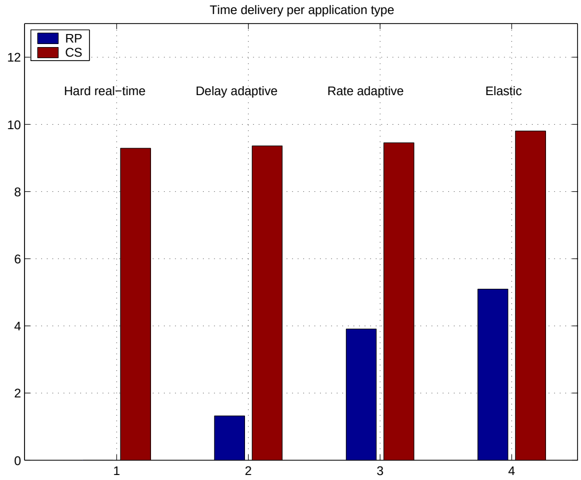

Figure 7.19: The time delivery for connections affected by failure per application type for two failed links at node with degree 4 , for dynamic (RP) and static (CS) bandwidth allocation.

$$\bar{K_j^s} = \frac{K_j}{|N|} \quad .$$

Similarly, the average number of connections at link j, with destination d, $\bar{K_j^d}$ is equal to

$$\bar{K_j^d} = \frac{K_j}{|N|} \quad .$$

The weighted utility of the request r_k is calculated at each link j as

$$u_k^{(j)} = \begin{cases} \frac{K_j^{s_k}+K_j^{d_k}}{\frac{2K_j}{|N|}} \cdot u_k, & (K_j^{s_k} + K_j^{d_k}) - \frac{2K_j}{|N|} > 0 \\ u_k, & (K_j^{s_k} + K_j^{d_k}) - \frac{2K_j}{|N|} < 0 \end{cases}$$

Results and Discussion

In this experiment we assume that the load in the network is *asymmetric*. We fix a node in the network and generate the traffic source destination pairs in such a way that there are p % connections with that node as an origin or destination. The performance of the network obviously depends on the degree of a selected node. We consider nodes degrees 2, 4 and 6 and percentages of asymmetry of 20 % and 40 %. The results are presented in Figures 7.20-7.23.

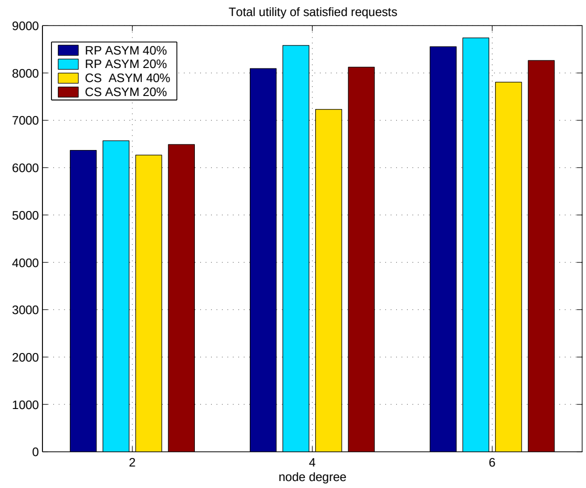

Figure 7.20: The total utility of satisfied requests as a function of node degree with asymmetric load in the network for dynamic (RP) and static (CS) bandwidth allocation.

From Figures 7.20-7.23, we can observe that, as the percentage of asymmetry is increasing, the performance of the network deteriorates. The number of unsatisfied requests is increasing, the traffic handling capacity is decreasing. Time delivery for connections affected by asymmetric load (including the selected node as a source or destination) is approaching 10 seconds when typical data connections and other connection-oriented data sessions are timed-out and disconnected. This deterioration in network performance is

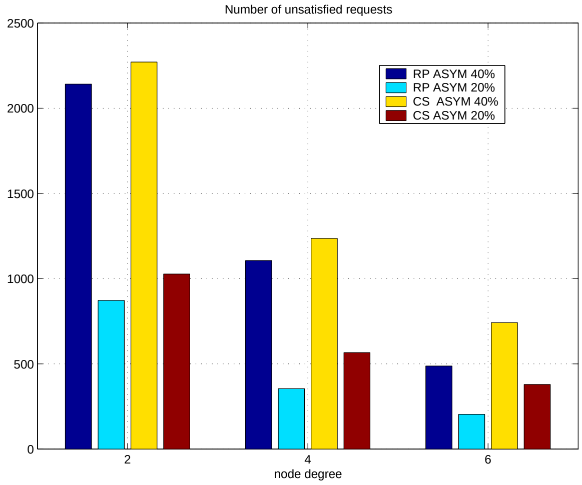

Figure 7.21: The number of unsatisfied requests as a function of node degree with asymmetric load in the network for dynamic (RP) and static (CS) bandwidth allocation.

greater as the node degree is smaller.

7.2.4 Varying path characteristics

Simulation Model and Experiments

The inference of network conditions such as link capacities, packet loss or propagation delays over links is of fundamental importance to a range of network aware applications. For example, the robust and efficient use of link capacities has always been a major issue and a crucial parameter in routing and traffic engineering, QoS management, streaming applications and in several other areas. Propagation delay is an important parameter for real-time applications such as voice and video, as the delay requirement of some applications will have impact on route selection.

In wireless networks the dynamics of the physical channel also affect efficient resource management aimed to achieve the classic trunking gains and to improve resource

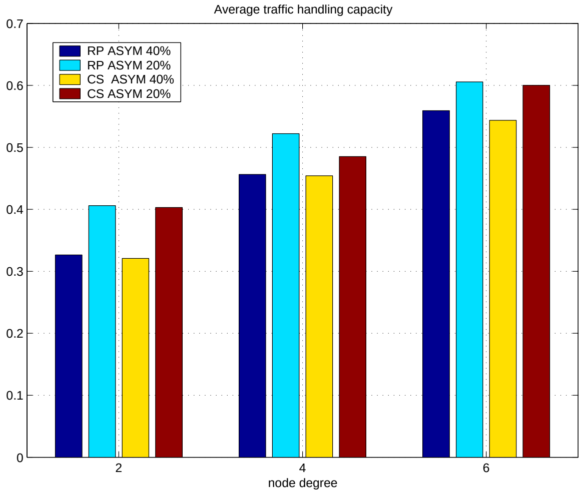

Figure 7.22: The average traffic handling capacity of connections affected by asymmetric load as a function of node degree with asymmetric load in the network for dynamic (RP) and static (CS) bandwidth allocation.

utilization. The dynamics of physical channels represent the inherent variability of wireless channels, which is both time-dependent and location-dependent. Wireless links experience two types of variations: slow variations (shadowing) and fast variations (fading). The duration of shadowing is in the order of seconds, while fading is in the order of milliseconds. In a mobile communication system, QoS guarantees have to be provided without knowing the future locations of the users. Therefore, the propagation conditions may vary significantly and achievable data rates may even become much lower than the specified maximum rates of the equipment [9, 61].

A n-state Markov chain model can be used to model link characteristics through time by using $n \times n$ transition probability matrix. We define the average state-holding times and the degradation ratios of link capacities D_m in the network for each state of the Markov chain. The weighted utility of the request is calculated at each link j as

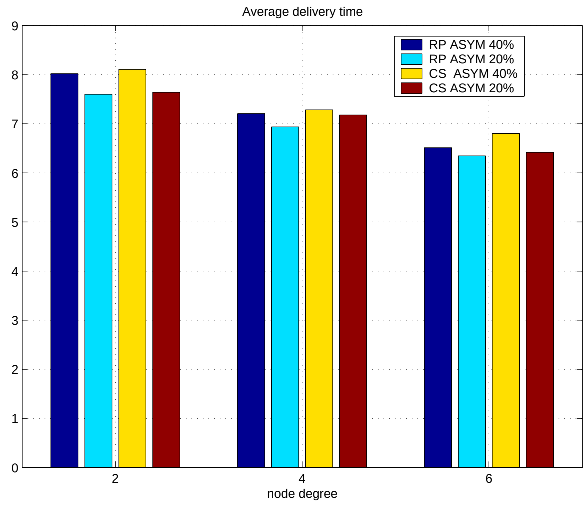

Figure 7.23: The average delivery time for connections affected by asymmetric load in the network as a function of node degree for dynamic (RP) and static (CS) bandwidth allocation.

$$u_k^{(j,m)} = (1 - D_{j,m}) \cdot u_k \tag{7.2}$$

where $0 \leq D_m \leq 1$.

A three-state Markov channel model is used to model time-varying links in the network. The transition probability matrix is shown in Equation 7.3. The average state-holding times are $t_1 = 13$, $t_2 = 15$, and $t_3 = 14$, and the degradation ratios are $D_1 = 0.4$, $D_2 = 0.2$, and $D_3 = 0$.

$$P = \begin{bmatrix} 0 & 0.8 & 0.2 \\ 0.5 & 0 & 0.5 \\ 0.2 & 0.8 & 0 \end{bmatrix} . \tag{7.3}$$

Results and Discussion

We model the links in the network according to the Markov channel model introduced above. We assume that the load is uniformly distributed in the network. The network performance metrics are represented in Figures 7.16-7.19.

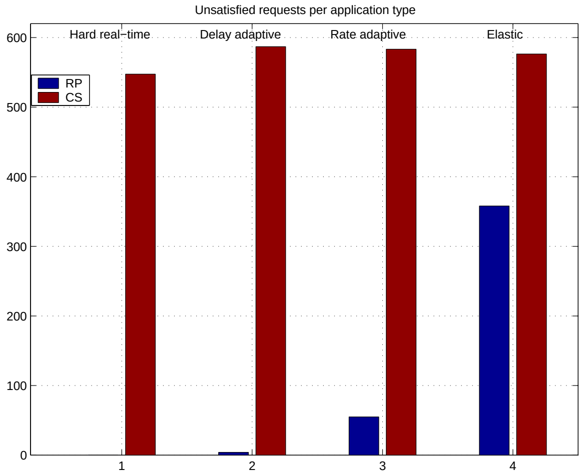

Figure 7.24: The number of unsatisfied requests per application type with time varying links, for dynamic (RP) and static (CS) bandwidth allocation.

Figure 7.24 shows the distribution of unsatisfied requests per application type with time varying links for dynamic and static bandwidth allocation. The advantage of dynamic bandwidth allocation becomes obvious from a revenue generation perspective, as more connection requests are satisfied than for the static bandwidth allocation policy. Dynamic bandwidth allocation allows a service provider to differentiate between different types of customers based on their priority and avoid disturbance of the high priority users in case of network resource fluctuations. It improves network traffic handling capacity and timely delivery of time critical applications and emergency response traffic.

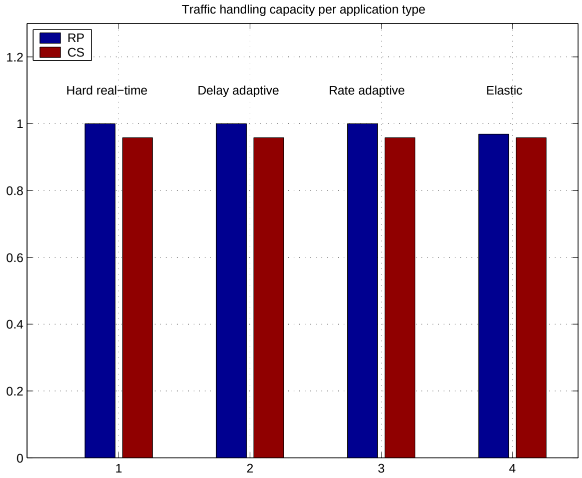

Figure 7.25: The traffic handling capacity of connections affected by link variations per application type, for dynamic (RP) and static (CS) bandwidth allocation.

7.3 Modelling Bandwidth Allocation Policies in Multiservice Networks

7.3.1 Single Link Model

Simulation Model and Experiments

In order to determine blocking probabilities and probabilities of preemption, for an initially empty system, we generate a sequence of events, arrival and departures of the connections, in a certain time interval. For those generated events we analyze the link occupancy and collect data on blocking probabilities for all connection types.

We analyze the system with three types of traffic and $n = 600$ servers. Traffic type 1 has a uniform resource request distribution with b_1 taking values between 4 and 8, and $\bar{b_1} = 6$. Traffic type 2 has a uniform resource request distribution with b_2 taking values between 6 and 12, and $\bar{b_2} = 9$. Traffic type 3 has a uniform resource request distribution

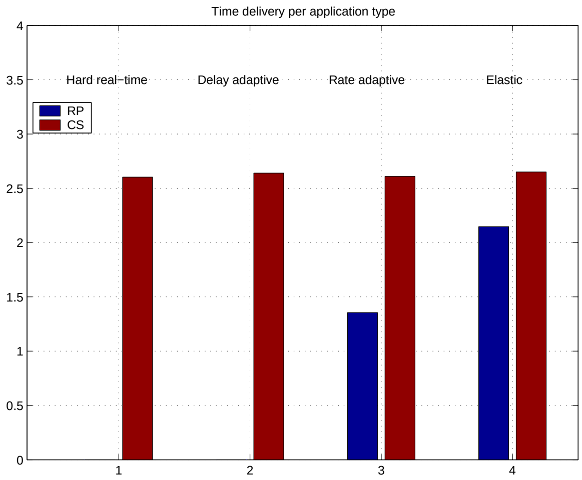

Figure 7.26: The time delivery for connections affected by link variations, for dynamic (RP) and static (CS) bandwidth allocation.

with b_3 taking values between 8 and 16, and $\bar{b_3} = 12$. Thus $\bar{b_1} \leq \bar{b_2} \leq \bar{b_3}$.

For a complete sharing policy when a new arrival is generated, we check whether sufficient resources are available to satisfy the request. If enough free resources are available, the requested resources are allocated and the connection is accepted on the link. Otherwise, the request is blocked.

For a guaranteed minimum policy, each traffic type has a guaranteed minimum of $C_r = 150$ servers which represents 25% of the total number of servers where a new connection arrival first tries to establish itself. If not successful, it tries to establish itself in the 150 servers shared with the blocked connections of other traffic types.

For preemptive bandwidth allocation, if not enough resources are available, an algorithm has to decide which ongoing connections of lower priority to preempt in order to establish the high-priority connection. We use the preemption algorithm proposed in [13] where an objective function can be adjusted by the service provider in order to stress the desired criteria for optimization or revenue optimization.

Results and Discussion

We present the results from analysis and simulation in this section. We fix the offered traffic of two classes of traffic and increase the offered traffic of the third traffic type and observe the blocking probabilities of the three traffic types in a lightly to highly loaded link. In the following figures we present results for blocking probabilities of each traffic type calculated from the model pb_r and obtained from the simulation pbs_r for complete sharing, guaranteed minimum and preemptive policy, respectively.

Figure 7.27 shows blocking probabilities obtained from the model presented in Section 6.1 and from the simulation of the system with a complete sharing policy. The figure shows results when we fix traffic of the two classes and increase traffic of the third class, $A3$, $A2$, $A1$, respectively. For the first case $A1 = 50$, $A2 = 25$ and $A3$ is increasing from 10 to 50 Erlangs. In the second case $A1 = 50$, $A3 = 25$ and $A2$ is increasing from 10 to 50 Erlangs. And in the third case $A2 = 50$, $A2 = 25$ and $A1$ is increasing from 10 to 50 Erlangs. Since in the complete sharing policy, connections that request a lower amount of bandwidth are more likely to be admitted, class 1 which requests the least amount of resources has the lowest blocking probability in all three cases. For all three cases, the blocking probabilities of all traffic types are increasing steadily as we increase the traffic of just one traffic type.

For those reasons, in order to protect some of the traffic we can implement the guaranteed minimum policy for which we present results on Figure 7.28 for the same three cases as for the complete sharing policy. For the guaranteed minimum policy, as we can see from the figure, the blocking probabilities of the "fixed" traffic types are increasing very slowly as we increase the offered traffic of the third traffic type. The reason for this slight increase is that the number of connections of the third traffic type, which are blocked in the guaranteed minimum of the resources, is increasing. In contrast, the number of connections blocked in the guaranteed minimum for two "fixed" traffic types remains constant, i.e., B_r' remains constant while B_r'' is increasing for the "fixed" traffic types.

Figure 7.29 presents results for the prioritized system in which traffic type 3 has the highest priority, type 2 medium priority and type 1 the lowest priority. Blocking probabilities are increasing as the traffic priority is decreasing and high-priority connections are less disturbed. Traffic type 3, as the highest priority traffic, has a non zero blocking probability only when there are not enough resources to accept all requests of that traffic type.

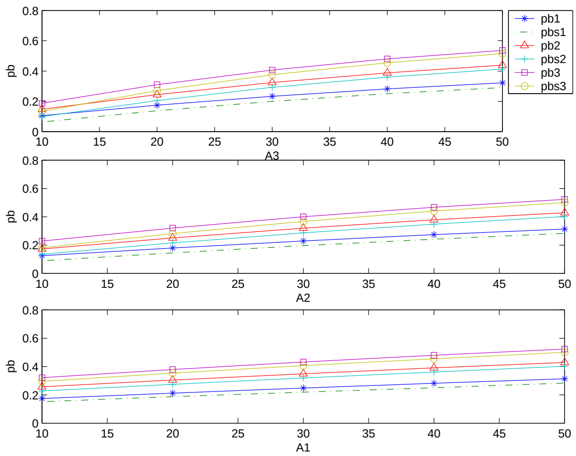

Figure 7.27: Blocking probabilities for CS policy as a function of traffic intensity.

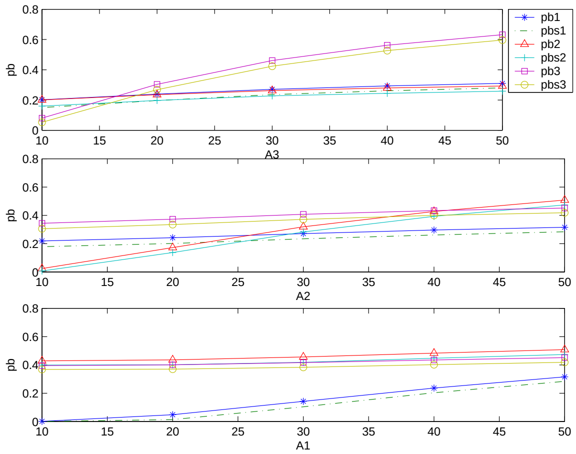

Figure 7.28: Blocking probabilities for GM policy as a function of traffic intensity.

7.3.2 Modelling of Utility based Bandwidth Allocation

Simulation Model and Experiments

In order to determine blocking probabilities and probabilities of preemption, for an initially empty system, we generate a sequence of events, arrival and departures of the connections, in a certain time interval. For those generated events we analyze the link occupancy and collect data on blocking probabilities for all connection types.

We analyze the system with four types of traffic and $n = 1000$ servers. Traffic type 1 has a uniform resource request distribution with b_1 taking values between 5 and 12, and $\bar{b_1} = 8.5$. Traffic type 2 has a uniform resource request distribution with b_2 taking values between 9 and 14, and $\bar{b_2} = 11.5$. Traffic type 3 has a uniform resource request distribution with b_3 taking values between 12 and 17, and $\bar{b_3} = 13.5$. Traffic type 4 has a uniform resource request distribution with b_3 taking values between 14 and 19, and $\bar{b_4} = 16.5$. Thus $\bar{b_1} < \bar{b_2} < \bar{b_3} < \bar{b_4}$.

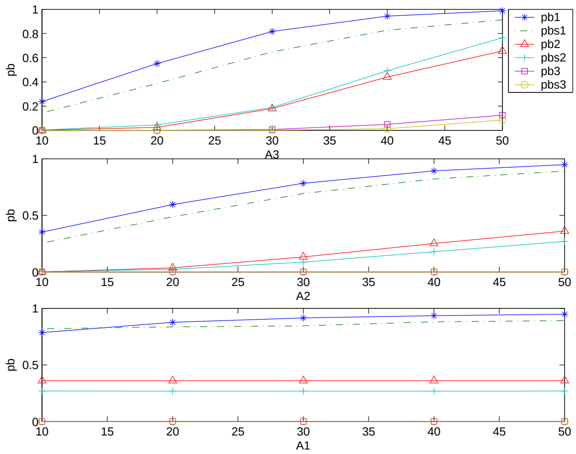

Figure 7.29: Blocking probabilities for Preemptive policy as a function of traffic intensity.

For a complete sharing policy, when a new arrival is generated, we check whether sufficient resources are available to satisfy the request. If enough free resources are available, the requested resources are allocated and the connection is accepted on the link. Otherwise, the request is blocked.

For a guaranteed minimum policy, each traffic type has a guaranteed minimum of $C_r = 200$ servers which represents 20% of the total number of servers where a new connection arrival first tries to establish itself. If not successful, it tries to establish itself in the 200 servers shared with the blocked connections of other traffic types.

For preemptive bandwidth allocation, if not enough resources are available, an algorithm has to decide which ongoing connections of lower priority to preempt in order to establish the high-priority connection. We use the preemption algorithm proposed in [13] where an objective function can be adjusted by the service provider in order to stress the desired criteria for optimization or revenue optimization. We analyze the system with four priority levels which are mapped from the connections utility value. Without loss of generality, we assume that utility function is a function of required bandwidth as represented on Figure 7.30. Since bandwidth is distributed uniformly, traffic type 1 will be mapped to all four priority levels equally, traffic type 2 will be mapped to priority levels 0, 1 and 2 equally, traffic type 3 will be mapped to priority levels 0 and 1 equally, while traffic type 4 will be entirely mapped to priority level 1 since all connections will have a utility value of 1 regardless of their bandwidth requirement.

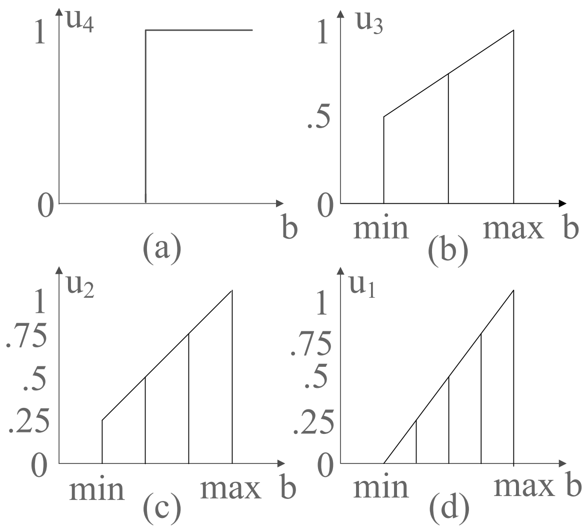

Figure 7.30: Utility (performance) of traffic type 4 (a), traffic type 3 (b), traffic type 2 (c) and traffic type 1 (d) as a function of bandwidth.

Results and Discussion

We present the results from analysis and simulation in this section. We fix the offered traffic of three classes of traffic and increase the offered traffic of the fourth traffic type and observe the blocking probabilities of the four traffic types in a lightly to highly loaded link. In the following figures we present results for blocking probabilities of each traffic type, $pb(r)$, and the total utility of all connections in the system $E[U]$ calculated from the model and obtained from the simulation, $pbs(r)$, $E[U]s$ for complete sharing, guaranteed minimum and preemptive policy.

Figures 7.31 and 7.32 show blocking probabilities obtained from the model presented in Section 6.1 and from the simulation of the system with a complete sharing policy. The figure shows results when we fix traffic for three classes and increase traffic of the fourth class. Three classes are fixed at 20 Erlangs and the fourth class is increasing from 20 to 60 Erlangs. Since in the complete sharing policy, connections that request a lower amount of bandwidth are more likely to be admitted, class 1 which requests the least amount of resources has the lowest blocking probability in all three cases. For all four cases, the blocking probabilities of all traffic types are increasing steadily as we increase the traffic of just one traffic type. Analytical results match the results obtained from the simulation of the system.

For those reasons, in order to protect some of the traffic we can implement the guaranteed minimum policy for which we present results on Figures 7.33 and 7.34 for the same four cases. For the guaranteed minimum policy, as we can see from the figure, the blocking probabilities of the “fixed” traffic types are increasing very slowly as we increase the offered traffic of the fourth traffic type. The reason for this slight increase is that the number of connections of the fourth traffic type, which are blocked in its guaranteed minimum of the resources, is increasing. In contrast, the number of connections blocked in the guaranteed minimum for three “fixed” traffic types remains constant, i.e., B'_r remains constant while B''_r is increasing for the “fixed” traffic types. Again, analytical results match the results obtained from the simulation of the system even for a highly loaded link and very large blocking probabilities.

Figures 7.35 and 7.36 present results for the prioritized system. Blocking probabilities of the traffic types with lower utility value are increasing as the traffic on the link is increasing for, as the high-priority connections are less disturbed. Traffic type 4, as the highest utility traffic, has a non zero blocking probability only when there are not enough resources to accept all requests of that traffic type.

On Figures 7.37, 7.38 and 7.39 we show the total utility of the connections on the link obtained from the model and from simulations $[EU_r]$ and $[EUs_r]$ as traffic type r increases and all other traffic types are fixed. As the offered traffic is increasing, total utility starts to increase but the gain in total utility of satisfied connections on the network becomes smaller for high traffic intensity as more and more connections are being blocked.

Tables 7.9-7.26 represent blocking probabilities on a 2-hop path according to the approximate model and simulation of such a path for traffic intensity of all source destination pairs and all traffic types equal to 10, 12.5 and 15 Erlangs. The model gives better results for lightly to moderately loaded links along the path.

7.3.3 Multihop path modelling

Simulation Model and Experiments

In order to determine blocking probabilities and probabilities of preemption, for an initially empty system, we generate a sequence of events, arrival and departures of the connections on the links in the path, in a certain time interval. For those generated events

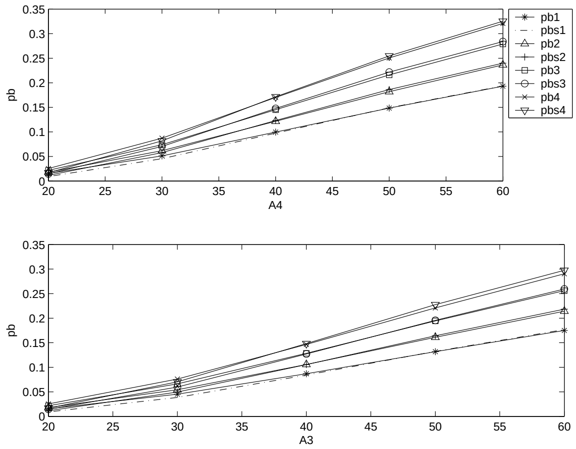

Figure 7.31: Blocking probabilities for the CS policy as a function of traffic intensity $A4$ and $A3$.

we analyze the links occupancy and collect data on blocking probabilities on individual links and a complete path for all connection types.

Results and Discussion

Tables 7.9-7.20 represent blocking probabilities on a 2-hop path according to the approximate model and simulation of such a path for traffic intensity of all source destination pairs and all traffic types equal to 10, 12.5 and 15 Erlangs. The model gives better results for lightly to moderately loaded links along the path.

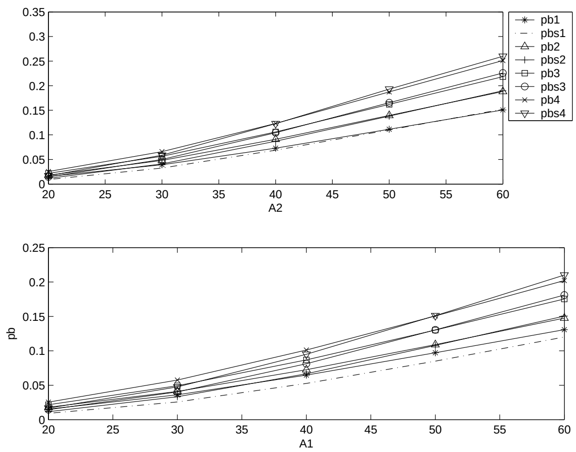

Figure 7.32: Blocking probabilities for the CS policy as a function of traffic intensity $A2$ and $A1$.

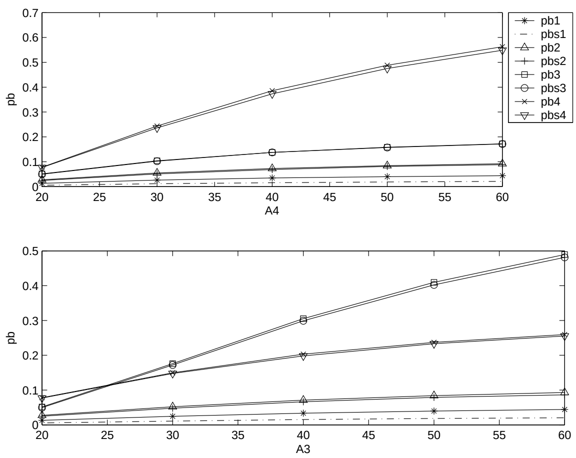

Figure 7.33: Blocking probabilities for the GM policy as a function of traffic intensity $A4$ and $A3$.

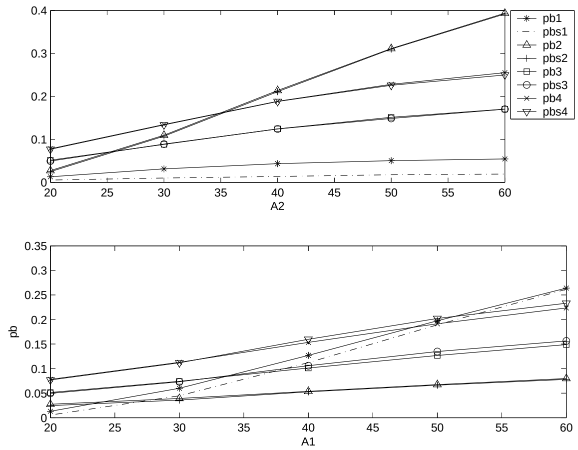

Figure 7.34: Blocking probabilities for the GM policy as a function of traffic intensity $A2$ and $A1$.

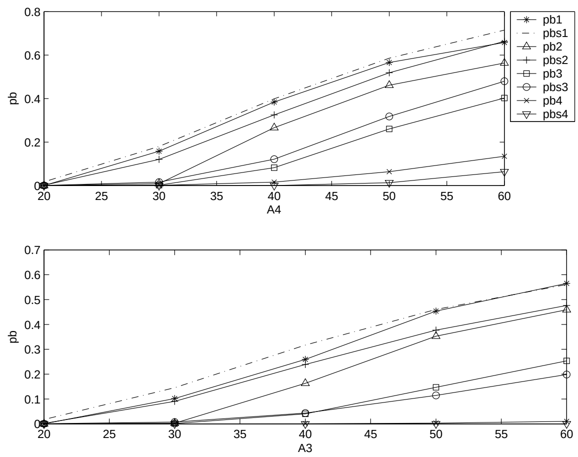

Figure 7.35: Blocking probabilities for the preemptive policy as a function of traffic intensity $A4$ and $A3$.

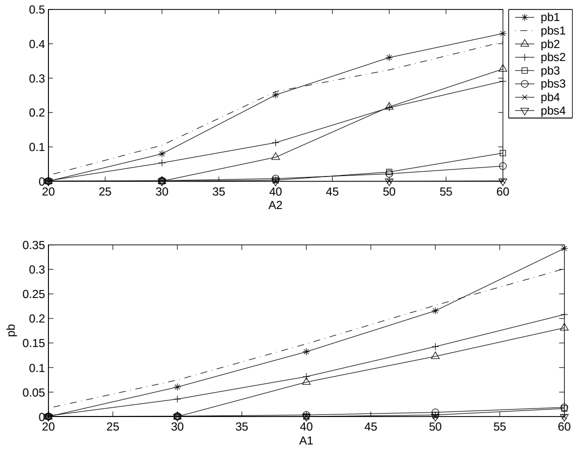

Figure 7.36: Blocking probabilities for the preemptive policy as a function of traffic intensity $A2$ and $A1$.

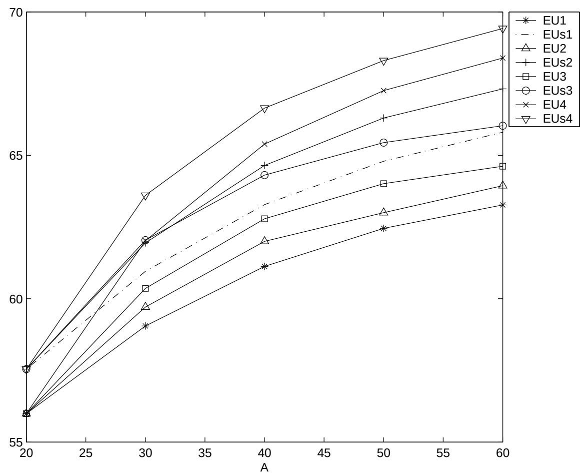

Figure 7.37: Total utility for the CS policy as a function of traffic intensity.

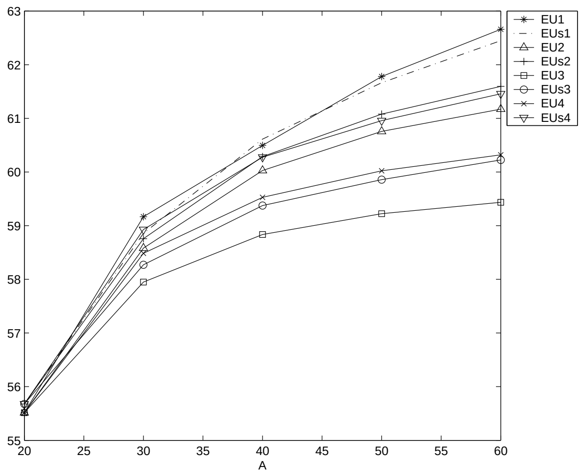

Figure 7.38: Total utility for the GM policy as a function of traffic intensity.

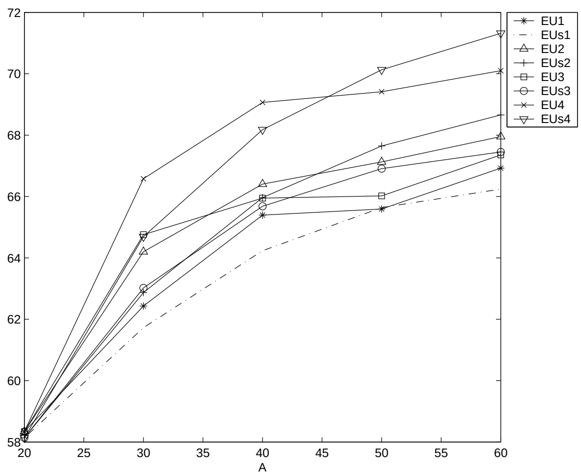

Figure 7.39: Total utility for the preemptive policy as a function of traffic intensity.

7.4 Chapter Summary

In this chapter, we have analyzed the preemption algorithms and compared to existing optimal algorithms for connection preemption through complexity calculations and a simulation study. Sub optimal preemption policies, based on random selection, due to their linear complexity, reduce dramatically the time to find a set of connections to be preempted. This is very important in the case of a large network with a large number of connections, given that connection preemption will need to be implemented as a real-time procedure. In a larger network, implementing a variation of QoS routing, connection preemption with random selection could provide a high quality of service to higher-priority network connections, while utilizing network bandwidth efficiently. We have also analyzed utility-based QoS model and a *generalized* bandwidth allocation scheme in overloaded network and for variable topology networks. It is clear that the dynamic bandwidth allocation scheme outperforms the static scheme in all aspects: total utility of satisfied requests, number of satisfied users, QoS guarantees, and network utilization. To validate our approximate analytical models we have developed a simulation of the system with implemented bandwidth allocation policies. We have shown that the analytical models presented in the paper can be effective in calculating approximate blocking probabilities of multiple traffic classes.

SD	pb1	pb2	pb3	pb4
01	0.008	0.009	0.011	0.013
02	0.015	0.018	0.021	0.025
12	0.008	0.009	0.011	0.013

Table 7.9: Source destination blocking probabilities for the CS policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.007	0.009	0.012	0.014
02	0.015	0.019	0.023	0.028
12	0.007	0.009	0.012	0.014

Table 7.10: Source destination blocking probabilities simulation results for the CS policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.007	0.007	0.0257	0.029
02	0.013	0.013	0.045	0.050
12	0.007	0.007	0.0257	0.039

Table 7.11: Source destination blocking probabilities for the GM policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.004	0.020	0.041	0.064
02	0.010	0.029	0.059	0.081
12	0.004	0.020	0.041	0.065

Table 7.12: Source destination blocking probabilities simulation results for the GM policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.014	0.009	0.001	0
02	0.032	0.018	0.001	0
12	0.013	0.008	0.001	0

Table 7.13: Source destination blocking probabilities for the preemptive policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.024	0.011	0.001	0
02	0.043	0.020	0.001	0
12	0.022	0.010	0.001	0

Table 7.14: Source destination blocking probabilities simulation results for the preemptive policy for traffic intensity A=10.

SD	pb1	pb2	pb3	pb4
01	0.040	0.049	0.058	0.068
02	0.073	0.087	0.104	0.120
12	0.034	0.041	0.049	0.056

Table 7.15: Source destination blocking probabilities for the CS policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.057	0.072	0.087	0.102
02	0.109	0.137	0.164	0.190
12	0.057	0.071	0.087	0.101

Table 7.16: Source destination blocking probabilities simulation results for the CS policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.030	0.054	0.086	0.119
02	0.054	0.096	0.150	0.201
12	0.026	0.047	0.076	0.107

Table 7.17: Source destination blocking probabilities for the GM policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.032	0.080	0.132	0.179
02	0.062	0.152	0.242	0.321
12	0.032	0.082	0.135	0.181

Table 7.18: Source destination blocking probabilities simulation results for the GM policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.104	0.055	0.01	0.000
02	0.168	0.091	0.021	0.001
12	0.080	0.040	0.01	0.000

Table 7.19: Source destination blocking probabilities for the preemptive policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.133	0.063	0.023	0.000
02	0.233	0.143	0.040	0.000
12	0.136	0.063	0.022	0.000

Table 7.20: Source destination blocking probabilities simulation results for the preemptive policy for traffic intensity A=12.5.

SD	pb1	pb2	pb3	pb4
01	0.082	0.100	0.118	0.136
02	0.142	0.171	0.199	0.226
12	0.071	0.086	0.105	0.122

Table 7.21: Source destination blocking probabilities for the CS policy for traffic intensity A=15.

SD	pb1	pb2	pb3	pb4
01	0.117	0.148	0.177	0.206
02	0.219	0.271	0.321	0.366
12	0.118	0.147	0.177	0.206

Table 7.22: Source destination blocking probabilities simulation results for the CS policy for traffic intensity A=15.

SD	pb1	pb2	pb3	pb4
01	0.059	0.100	0.145	0.186
02	0.105	0.172	0.241	0.301
12	0.052	0.089	0.131	0.174

Table 7.23: Source destination blocking probabilities for the GM policy for traffic intensity A=15.

SD	pb1	pb2	pb3	pb4
01	0.079	0.152	0.220	0.279
02	0.146	0.276	0.383	0.473
12	0.079	0.155	0.221	0.281

Table 7.24: Source destination blocking probabilities simulation results for the GM policy for traffic intensity A=15.

SD	pb1	pb2	pb3	pb4
01	0.257	0.212	0.096	0.006
02	0.407	0.331	0.150	0.010
12	0.272	0.192	0.066	0.004

Table 7.25: Source destination blocking probabilities for preemptive policy for traffic intensity A=15.

SD	pb1	pb2	pb3	pb4
01	0.352	0.271	0.129	0.001
02	0.461	0.377	0.204	0.001
12	0.352	0.271	0.130	0.001

Table 7.26: Source destination blocking probabilities simulation results for preemptive policy for traffic intensity A=15.

Chapter 8

Summary and Future Work

This chapter summarizes the work and suggests extensions for future work.

8.1 Summary

As the networks are evolving to multiservice networks, supporting heterogeneous traffic mix with varying traffic characteristics, QoS constraints and bandwidth requirements, service providers must be capable to support the needs of users who have urgent needs for communications resources or wish to schedule QoS enabled sessions in advance and maintain a desired level of QoS throughout the session without interruptions. We have analyzed both static and dynamic resource allocation, and compared them through complexity calculations and a simulation study. We have introduced a utility-based QoS model to describe users' value which account users' QoS requirements. Our schemes allow a service provider to differentiate between different types of customers based on their priority or the service charges that they pay, in order to offer real time services, to provide QoS guarantees for multimedia traffic and to guarantee stability even in overloaded conditions. We have presented an analysis of calculating blocking probabilities in multi rate multi class systems where the requests from the same class do not have the same resource requirements. In order to calculate the blocking probabilities for various bandwidth allocation policies, we have expanded existing multi rate Erlang B and $M/M/c/c$ models. We considered generalizations of loss networks allowing the CS and GM sharing policies as well as a prioritized preemptive policy. To validate our approximate analytical models we have developed a simulation of the system with implemented bandwidth allocation policies. We have shown that the analytical models presented here can be effective in calculating approximate blocking probabilities of multiple traffic classes. Due to the generality of our framework and the bandwidth allocation scheme, they can be applied to different networks, such as wireless and ad-hoc networks, with heterogeneous traffic mixes and varying traffic characteristics, QoS constraints and bandwidth requirements.

8.2 Future Work

Next generation networks will offer various levels of service with prices associated with each type of service. The pricing structure will directly influence the load present in the network as well as its performance, thus playing an important role in *commercial network dimensioning.* For example, market based network control will certainly be based on dynamic resource re-allocation and possibly, *preemption.* And users who pay more (or who are designated "high priority" in a public interest sense) should be offered more reliable services [12]. For those reasons, one area for future work would be to study economic aspect and benefits of using various bandwidth allocation from network utilization and revenue generation perspectives for different pricing/charging policies.

An important objective of contemporary and future Internet traffic engineering, is to support *reliable network operations* by providing mechanisms that enhance network integrity and by adopting policies that accommodate network survivability. The network needs to assure that high priority requests can be always routed through relatively favorable paths within a differentiated services environment. Due to statistical fluctuations, user mobility or occasional unavailability of resources (due to faults or attacks), some connections that could otherwise have been accepted if the traffic load were better balanced (with consequently larger revenue to the network), are instead rejected. For lower priority connections, a preemption policy complemented with an adaptive rate scheme could be more efficient. Some applications such as non-real-time video or WWW traffic, that are designed to adapt to transmissions with less than the full desired bandwidth, should be assigned to lower priority levels. By reducing their rate, those connections would not be torn down, there would not be service disruption, extra setup and tear down signaling, or rerouting. In the future, whenever there exists available bandwidth in the network, the reduced-rate connections would increase their rate to the original reserved bandwidth.

Prioritized resource allocation can be used for reestablishing connections that have been disrupted by a failure. One aspect of fault recovery of the connection affected by the failure is the procedure for their re-acceptance into the network. Since the failure typically results in several nodes being sources for affected connections, in each of those nodes there will be many connections to simultaneously restore. Restoration algorithm needs to determine which connections are to be restored for finding the best combination of which connections would be rerouted from the set of connections affected by failure. Objectives of the restoration algorithms can be seen as the inverse of the objectives of the rerouting algorithms.

Chapter 9

Publications

- V. Stanisic and M. Devetsikiotis, “An Analysis of Bandwidth Allocation Strategies in Multiservice Networks”, IEEE ICC 2005, Seoul, Korea, may 2005.
- V. Stanisic and M. Devetsikiotis, “Dynamic Utility-based Bandwidth Allocation Policies: The Case of Overloaded Network” IEEE ICC 2004, Paris, France, June 2004.
- V. Stanisic and M. Devetsikiotis, “A Dynamic Study of Providing Quality of Service Using Preemption Policies with Random Selection,” IEEE ICC 2003, Anchorage, USA, May 2003.
- V. Stanisic and M. Devetsikiotis, “A Framework for Providing Quality of Service Using Preemption Policies” CACC Technical report, November 2002.
- V. Stanisic and M. Devetsikiotis, “Traffic Restoration Algorithms in Communication Networks” accepted in WSEAS 2004, Athens, Greece, July 2004.
- V. Stanisic and M. Devetsikiotis, “Application Based Resource Allocation Policies in

a Network with Heterogeneous Users", submitted to the Computer Communications Journal, January 2005.

Bibliography

[1] H. Ahmadi, J. Chen, and R. Guerin. Dynamic Routing and Call Control in High-speed Networks. In *Teletraffic and Data Traffic: Socio-Economic Aspects*, pages 397–403, June 1991. Proc. 13th International Teletraffic Congress (ITC-13), Copenhagen, Denmark.

[2] D. Awduche, L. Berger, and D. Gan et al. *Extensions to RSVP for LSP Tunnels*. IETF, RFC 3209, December 2001.

[3] D. Awduche, J. Malcolmet, and J. Agogbua et al. *Requirements for Traffic Engineering over MPLS*. IETF, RFC 2702, September 1999.

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

[5] C. Beard and H. Folts. *Requirements for Emergency Telecommunication Capabilities in the Internet*. IETF,Internet draft, draft-ietf-ieprep-requirements-00.txt, June 2002.

[6] C. Beard and V. S. Frost. Prioritized resource allocation for stressed networks. *IEEE/ACM Transactions on Networking*, 6(5):618–633, October 2001.

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

[8] S. Borst and D. Mitra. Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic. *IEEE Journal on Selected Areas in Communications*, 16(5), June 1998.

[9] Y. Cao and V. O. K. Li. Utility-oriented Adaptive QoS and Bandwidth Allocation in Wireless Networks. In *ICC 2002*, New York City, April 2002.

[10] G. L. Choudhury, K. K. Leung, and W. Whitt. An algorithm to compute blocking probabilities in mult-rate multi-class multi-resource loss models. *Advances in Applied Probability*, 27:1104–1143, 1995.

[11] G. L. Choudhury, K. K. Leung, and W. Whitt. Efficiently Providing Multiple Grades of Service with Protection Against Overloads in Shared Resources. *AT&T Technical Journal*, 74(4):50–63, 1995.

[12] L. DaSilva, D. Petr, and N. Akar. Equilibrium Pricing in Multiservice Priority Based Networks. In *Globecom 97*, Phoenix, AZ, November 1997.

[13] J.C. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl. A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting. In *INFOCOM 2002*, New York City, June 2002.

[14] P. Dharwadkar, H. J. Siegel, and E. K. P. Chong. A Study of Dynamic Bandwidth Allocation with Preemption and Degradation for Prioritized Requests. Technical Report TR-ECE 00-9, Purdue University, 2000.

[15] P. Dharwadkar, H. J. Siegel, and E. K. P. Chong. A Heuristic for Dynamic Bandwidth Allocation with Preemption and Degradation for Prioritized Requests. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, pages 547–556, Phoenix, Arizona, April 16–19 2001.

[16] N. G. Duffield, P. Goyal, A. G. Greenberg, P. P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive. A Flexible Model for Resource Management in Virtual Private Networks. In *SIGCOMM*, pages 95–108, 1999.

[17] Anwar I. Elwalid and Debasis Mitra. Effective Bandwidth of General Markvian Traffic Sources and Admission Control of High Speed Networks. *IEEE/ACM Transactions on Networking (TON)*, 1(3):329–343, 1993.

[18] G. Karagiannis et al. A framework for QoS and mobility in the internet next generation. Ericsson Open Report, September 2000.

[19] G.J. Foschini et al. Optimum allocation of servers to two types of competing customers. *IEEE Transactions on Communications*, 29(7):1051–1055, July 1981.

[20] T. Egawa et al. Qos restoration for dependable networks. Network Operations and Management Symposium, NOMS'98, February 1998. New Orleans, Louisiana.

[21] K. Fall. Simulating the Internet: challenges and methods. IX Computer Science Symposium, Mayab University, Yucatan,Mexico, April 1998.

[22] F. Le Faucheur and W. Lai. *Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering.* IETF, RFC 3564, July 2003.

[23] N.L.S. Fonseca and R.A. Facanha. Statistical Multiplexing of Self-Similar Sources. In *Proceedings of IEEE Globecom 00*, volume 3, pages 1334 –1338, San Francisco, Nov.-1 Dec. 2000 2000.

[24] G. Gopal, C. Kim, and A. Weittrib. Algorithms for Reconfigurable Networks. pages 341–347, 1991. Proc. 13th International Teletraffic Congress (ITC-13), Copenhagen, Denmark.

[25] I. Gopal and T. Stern. Optimal call blocking policies in an integrated services environment. In *Proceedings of the 17th Conference on Information Science and Systems*, pages 383–388. The Johns Hopkins University, Baltimore, MD, 1983.

[26] Wayne D. Grover. *Mesh-based Survivable Networks:Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall PTR, New Jersey, 2003.

[27] L. Gun and R. Guerin. Bandwidth Management and Congestion Control Framework of the Broadband Network Architecture. *Computer Networks and ISDN Syst.*, 26(1):61–78, 1993.

[28] B. Jamoussi, L. Andersson, and R. Callon et al. *Constraint-Based LSP Setup using LDP*. IETF, RFC 3212, January 2002.

[29] S. Jordan and P. Varaiya. Control of multiple service, multiple resource communication networks. *IEEE Transactions on Communications*, 42(11):2979–2988, November 1994.

[30] F. Kamoun and L. Kleinrock. Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions. *IEEE Transactions on Communications*, COM-28:992–1003, 1980.

[31] J. Kaufman. Blocking in a shared resource environment. *IEEE Transactions on Communications*, 29(10):1474–1481, November 1981.

[32] F.P. Kelly. Charging and Accounting for Bursty Connections. In J. Surdej, D. Fraipont-Caro, E. Gosset, S. Refsdal, and M. Remy, editors, *Internet Economics*, pages 253–278, 1997.

[33] P. B. Key. Optimal control and trunk reservation in loss networks. *Probability in the Engineering and Informational Sciences*, (4):203–242, 1990.

[34] B. Kraimeche and M. Schwartz. Circuit access control strategies in integrated digital networks. In *INFOCOM 1984*, pages 230–235, San Francisco, April 1984.

[35] Richard J. La and Venkat Anantharam. Utility-Based Rate Control in the Internet for Elastic Traffic. *IEEE/ACM Transactions on Networking (TON)*, 10(2), 2002.

[36] C. Laws. On Trunk Reservation in Loss Networks. In F.P. Kelly and R.J. Williams, editors, Stochastic Networks, pages 187–198. Springer-Verlag, 1995.

[37] S. A. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23:686–710, 1975.

[38] W.P. Lovegrove and J.L. Hammond. Simulation methods for studying nonstationary behaviour of computer networks. *IEEE Journal on Selected Areas in Communications*, 8(9):1696–1708, December 1990.

[39] S. Maniatis, E. G. Nikolouzou, and I. S. Venieris. QoS Issues in the Converged 3G

Wireless and Wired Network. *IEEE Communications Magazine*, 40(8):44–53, August 2002.

[40] S. Nakagawa, Y. Kato, S. Nakai, K. Ogura, and H. Saito. Self-sizing Network Operation systems in ATM networks. pages 563–572, 1996. IEEE/IFIP 1996 Network Operation and Management Symposium (NOMS'96), Kobe, Japan.

[41] V. Nguyen. On the optimality of trunk reservation in overflow processes. *Probability in the Engineering and Informational Sciences*, (5):369–390, 1991.

[42] A. A. Nilsson, M. Perry, A. Gersht, and V. B. Iversen. On Multi-rate Erlang-B Computations. In *Proceedings of 16th International Teletraffic Congress*, pages 1051–60, Edinburgh, June 1999.

[43] M. Peyravian and A. D. Kshemkalyani. Decentralized Network Connection Preemption Algorithms. *Computer Networks and ISDN Syst.*, 30(11):1029–1043, June 1998.

[44] Jingyu Qiu and Edward W. Knightly. Measurement-Based Admission Control with Aggregate Traffic Envelopes. *IEEE/ACM Transactions on Networking*, 9(2):199–210, 2001.

[45] M. I. Reiman. Optimal trunk reservation for a critically loaded link. In *In Proceedings 13th International Teletraffic Congress*, 1991.

[46] J. Roberts. A service system with heterogeneous user requirements. *Performance of Data Communications Systems and their Applications*, pages 423–431, 1981.

[47] K.W. Ross and D.H.K. Tsang. The stochastic knapsack problem. *IEEE Transactions on Communications*, 37(7):740–747, July 1989.

[48] C. Saraydar, N. Mandayam, and D. Goodman. Efficient Power Control Via Pricing in Wireless Data Networks. *IEEE Transactions on Communications*, 50(2):291–303, 2002.

[49] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communication*, 13(7):1176–1188, September 1995.

[50] Lei Song and Narayan B. Mandayam. Hierarchical SIR and Rate Control on the Forward Link for CDMA Data Users under Delay and Error Constraints. *IEEE Journal on Selected Areas in Communications*, 19(10):1871–1882, 2001.

[51] S.Shekhar, K. Balaji, V. Venkatesh, C. Beard, D. Medhi, A. V. Liefvoort, W. Alanqar, and A.Nagarajan. A Case Study on Evaluating the Benefits of MPLS Traffic Engineering through Constraint-Based Routing and Network Controls. In *ICC 2002*, New York City, May 2002.

[52] V. Stanisic and M. Devetsikiotis. A Framework for Providing Quality of Service Using Preemption Policies. CACC Technical report, November 2002.

[53] V. Stanisic and M. Devetsikiotis. A Dynamic Study of Providing Quality of Service Using Preemption Policies with Random Selection. In *IEEE ICC 2003*, Anchorage, Alaska, May 2003.

[54] V. Stanisic and M. Devetsikiotis. Dynamic Utility-based Bandwidth Allocation Policies:

The Case of Overloaded Network. In *Proceedings of IEEE ICC 2004*, Anchorage, Alaska, May 2004.

[55] V. Stanisic and M. Devetsikiotis. An Analysis of Bandwidth Allocation Strategies in Multiservice Networks. In *Proceedings of IEEE ICC 2005*, Seoul, Korea, May 2005.

[56] V. Stanisic and M. Devetsikiotis. Resource Allocation Policies in Access Networks with Heterogeneous Users. submitted to Computer Communication, January 2005.

[57] W. Sun, P. Bhaniramka, and R. Jain. Quality of Service Using Traffic Engineering over MPLS: An Analysis. 25th Annual IEEE Conference on Local Computer Networks, November 2000. Tampa, Florida.

[58] R.E. Walpole and R.H. Myers. *Probability and statistics for engineers and scientists.* Macmillan, New York, 1985.

[59] H. Xiao and C. Beard. Support for high priority traffic using preemption. Technical report, University of Missouri-Kansas City, 2000. technical report.

[60] M. Yoo, C. Qiao, and S.Dixit. QoS Performance of Optical Burst Switching in IP-Over-WDM Networks. *IEEE Journal on Selected Areas in Communications*, 18(10), October 2000.

[61] Jens Zander. Affordable QoS in Future Wireless Networks Myth or Reality? 2nd International Workshop on Quality of future Internet Services, September 2001.