Dynamic bandwidth allocation in multi-class IP networks using utility functions.

Rakocevic, Veselin

Dynamic Bandwidth Allocation in Multi-Class IP Networks using Utility Functions

Veselin Rakočević

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Department of Electronic Engineering
Queen Mary, University of London

January 2002

to Svetlana and Vera

ABSTRACT

The Internet needs to evolve from a single-service data network into the multiservice intelligent IP-based network capable of satisfying diverse performance requirements. The way network resources, primarily bandwidth, will be allocated in the future network presents a very important research issue. Bandwidth allocation should be simple to implement and optimally designed to maximise the network performance. The majority of the research on bandwidth allocation in the Internet deals with the problem of bandwidth allocation in a single-service environment, in which the entire Internet traffic is treated as data transfer.

This Thesis presents **Dynamic Bandwidth Partitioning**, a new bandwidth allocation scheme for the multiservice Internet. In this scheme, the link bandwidth is partitioned in order to isolate the different traffic types and at the same time maximise the performance. The partitioning is dynamic; it changes according to a simple ‘additive increase, additive decrease’ linear control algorithm. The information about the change in the partitioning comes from the measured level of network performance. In other words, the Dynamic Bandwidth Partitioning scheme is user-oriented, adaptive and reasonably simple to implement.

The contribution of this work is in the introduction of the new bandwidth allocation scheme, and in the metric that was used for the network performance evaluation. This new metric is called **connection utility**. Connection utility is the assessment of the quality of service level which a user of an Internet application derives from the network performance. It is measured by using *utility functions.* The significant novelty of this work is in including *non-concave* utility functions for real-time traffic classes. Based on the information about the utility of the active traffic on the link, a decision about changes in the bandwidth allocation is made.

This Thesis presents the Dynamic Bandwidth Partitioning scheme in detail, including the partitioning algorithm and different utility functions that were used. Through extensive simulation the scheme is compared to several other bandwidth allocation concepts. The results presented here clearly show the network environments in which implementation of the new scheme can prove to be very efficient. Furthermore, the Thesis presents the guidelines for the implementation of the scheme in the MPLS-capable networks.

ACKNOWLEDGMENT

Many people have contributed to this Thesis in different ways. I think about all of them with joy at the time of finalisation of the Thesis, and find it very difficult not to mention everyone.

This research has been sponsored by Fujitsu Telecommunications Europe Ltd. I am grateful to FTEL for their sponsorship and for the technical support. I would like to especially thank Mr. Graham Cope from FTEL for numerous technical discussions and valuable advices.

I would like to thank everybody in the Department of Electronic Engineering at Queen Mary, University of London for creating and maintaining a nice and relaxing working environment. My special gratitude goes to Prof. John Griffiths, Dr. John Schormans and Prof. Laurie Cuthbert for their continuous support during my research.

The finalisation of this Thesis would not be possible without the love and support from my family, especially my parents Pero and Agnes and my sister Maja. I am grateful to them for the continuous encouragement and support during the years of my study in London.

Finally, I would like to thank my dear colleagues and friends whom I've met at Queen Mary over the last three years: Arif Al-Hammadi, Husam Awadalla, Eliane Bodanese, Kok Ho Huen, Andy Martin, Maurizio Migliozzi, Robert Stewart and Tijana Timotijevic for their help and friendship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BA	Behaviour Aggregate
BGP	Border Gateway Protocol
CR-LDP	Constraint-based Routed LDP
DBP	Dynamic Bandwidth Partitioning
DiffServ	Differentiated Services
DS	Differentiated Services
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
FIFO	First In First Out
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
LAN	Local Area Network
LDP	Label Distribution Protocol
LIFO	Last In First Out
LSP	Label Switch Path
LSPID	LSP Identifier
LSR	Label Switch Router
MPLS	Multiprotocol Label Switching
MTU	Maximum Transfer Unit
NHLFE	Next Hop Label Forwarding Entry
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PHB	Per-Hop Behaviour
QoS	Quality of Service
RED	Random Early Detection

RFC	Request For Comments
RSVP	Resource Reservation Protocol
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TLV	Type-Length-Value
TOS	Type of Service
TTL	Time To Live
UDP	User Datagram Protocol
VCI	Virtual Circuit Indentifier
VPI	Virtual Path Identifier
VPN	Virtual Private Network
WDM	Wavelength Division Multiplex

CHAPTER *1*

INTRODUCTION

1.1. Overview

In the last couple of years, we have witnessed an exponential rise in the use of Internet, both in the number of users, and in the number of various applications these users need. Connectionless IP-based 'best-effort' forwarding of Internet traffic, which was characteristic for the first decades of communication networks, was very efficient for the data transfer applications. The only performance issue for the best-effort traffic forwarding was *fairness*, which usually indicated the equal distribution of the bottleneck bandwidth in the network. In the new application environment the necessity for new ideas for the traffic forwarding has become apparent.

New sophisticated real-time applications include video conferencing, video on demand, distance learning, distributed video gaming, etc. These applications require better and more reliable network performance. They demand firm *guarantees* from the network that certain resources (bandwidth, buffer space) will be reserved for them. Furthermore, the end-users' requirements are higher now, in terms of performance, accessibility and security. The optimal network of the future will not be the one which provides *equal* allocation of its resources to all active traffic connections, but the one that provides Quality of Service guarantees to the highest number of end-users.

Therefore, in the new multi-class Internet the network designers are facing a complicated problem of optimising the network control to satisfy both the issue of *fairness* for the data traffic and the issue of *performance guarantees* for the real-time traffic.

This Thesis analyses the necessity for the quality of service guarantees in the new Internet environment. The problem that is particularly analysed here is the problem of resource allocation in the multi-class Internet. The Thesis presents a new resource allocation scheme which is adaptive and user-oriented, designed to provide the guarantees to the traffic flows that require them.

A lot of work has been done concerning the issues of resource allocation and fairness in a single-service Internet environment, for the network serving a single type of elastic data traffic. The objective there was to use all available bandwidth while trying to achieve some *fairness* in the way the bandwidth is shared between different traffic flows. In the multi-class Internet, fairness needs to be observed in a different way. The traffic is now differentiated and has very diverse performance requirements. We need a universal method of assessing the network performance, which needs not to be purely the amount of resources that is given to a network connection, but rather the end-users' *benefit* of using a connection that has been given a certain amount of resources.

This Thesis analyses strict *partitioning* of network resources as a means to obtain the required network performance guarantees. Although network resources are numerous, the focus in this work is only on *bandwidth*. Bandwidth partitioning makes traffic from different traffic classes independent, and therefore protects high-priority traffic from the effect of sudden burstiness of the low-priority traffic. The concept of bandwidth partitioning has been analysed previously [NUN99][GUP95][ROS95], with a common conclusion that it is ineffective, due to low network utilisation. It is considered to perform better then the conventional best-effort scheme only in the environment of very high traffic loads.

This Thesis presents a novel approach to bandwidth partitioning, showing that an application-oriented and adaptive bandwidth partitioning scheme can perform better then the best-effort scheme even for moderate traffic loads.

The aim of research presented here is to prove that a simple bandwidth allocation control is more efficient in a multi-class network environment then the conventional accept-all policy of equal treatment of diverse network connections. The objectives of the research are summarised below:

- to define and analyse all aspects of the new bandwidth allocation scheme
- by experimenting with different parameters, to find the optimal design for the new scheme
- by comparing the scheme performance with other bandwidth allocation concepts, to define and discuss the network and traffic environments for which the new scheme would be optimal

The following two sections will present in more detail the contribution of this research and the structure of the Thesis.

1.2. Contribution of the Thesis

The main contribution of the research presented in this Thesis is the detailed definition of **Dynamic Bandwidth Partitioning,** a new bandwidth allocation scheme for the multi-class Internet.

The scheme takes under consideration the diversity of performance requirements that are put on the network. The available link bandwidth is partitioned between a number of traffic classes, so that the forwarding of the traffic classes is independent. By using a simple algorithm, the scheme measures the network performance on the link, and updates the way the link bandwidth has been partitioned, with the objective to increase the average performance level.

The second important contribution is the new performance metric that is being used.

Each user of an Internet application derives a certain **utility** from the network performance. The end-users of an Internet application are not interested in the amount of network bandwidth that is available to them, but rather in what they can obtain from that amount of bandwidth. It should be important to us how *satisfied* will the end-users be with the network performance. We model the end-users' satisfaction with the network performance by using **utility functions**. Each end-user application has a utility function defined. This function relates the allocated bandwidth to the user satisfaction, rating that satisfaction on a scale from 0 to 1.

Utility functions have greater significance in the multi-class network environment. The current Internet assumes that the utility function that drives bandwidth allocation is uniform among the users. However, this is not true in the multi-class network. The utility of a service is flexible according to user's subjective perceptions, and to the requirements of the application. It is even possible to define a different utility function to each user of each Internet application. The precise definition of the utility functions is therefore a very complicated problem. That is why we make an approximation in this research by defining a finite number of traffic classes, and defining a single utility function per each traffic class. This Thesis will show that by using well-defined utility functions it is possible to efficiently evaluate the network performance.

The new performance measure that is used in this research is the *connection utility.* The connection utility of each traffic connection is a mean value of the utility generated while that connection was active in the network. The level of the mean connection utility on the network link is used in the Dynamic Bandwidth Partitioning scheme as a feedback information for making the decisions on the partitioning updates.

In summary, the Dynamic Bandwidth Partitioning scheme is

- **user-oriented**, since end-users' *utility* is used as the feedback information for bandwidth allocation updates
- **dynamic**, since the changes in the utility are closely monitored and appropriate actions in the bandwidth partitioning are performed

- **QoS-aware,** since the available bandwidth is partitioned between different traffic classes; admission control is introduced, and minimal bandwidth levels are guaranteed
- **easy to implement,** since a simple additive increase, additive decrease linear control scheme is used, similar to the scheme used in TCP congestion control

The details of each of these issues will be explained in more detail in the Thesis.

During the research, a network simulator was specially developed in order to evaluate the performance of the scheme. The simulator is written in C, and is able to measure the connection utility and a number of conventional performance measures based on which the comparison of Dynamic Bandwidth Partitioning scheme with other bandwidth allocation concepts has been done.

The Dynamic Bandwidth Partitioning scheme, its evaluation and the implementation issues have been presented in a number of conference publications. The list of author's publications is given in Appendix A.

1.3. Structure of the Thesis

Chapter 2 analyses the quality of service issue in the Internet. We start by trying to define the quality of service, both from the users' perspective and from the providers' perspective. The first detailed explanation of the utility is given in this Chapter, and the difference between utility and *utilisation* is underlined. After that, the Internet infrastructure is presented through the analysis of the Internet Protocol (IP) and the Transmission Control Protocol (TCP). The Chapter is concluded with the overview of multiservice QoS-aware architectures and techniques currently available: Integrated Services with RSVP, Differentiated Services, and MPLS.

Chapter 3 analyses the problem of bandwidth allocation in the multi-class Internet. The issues of fairness, congestion control and different bandwidth allocation concepts are presented here. This Chapter presents the research that has been done so far in optimising the bandwidth allocation. It introduces the concept of bandwidth partitioning, and presents the conclusions of the previous research on the efficiency of such a concept.

Chapter 4 presents the Dynamic Bandwidth Partitioning scheme in detail. The scheme uses utility as an optimisation parameter and at the beginning of this Chapter an overview of similar optimisation approaches is given. Then each of the aspects of the scheme, namely bandwidth allocation, admission control and partitioning algorithm are explained. The Chapter is concluded with an analysis of the implementation issues, which includes the introduction of a new resource management scheme in the Virtual Private Networks, the *Hierarchical* Dynamic Bandwidth Partitioning.

Chapter 5 presents the simulation model. Specially built simulator was designed for this research. The simulator has been written in C programming language. This Chapter presents the network and traffic model, including the detailed analysis of the utility functions that were used, with the discussion of other utility functions that could have been used to model the observed traffic classes. Furthermore, this Chapter provides the mathematical analysis used for the validation of the simulator.

Chapter 6 presents the results of numerous experiments that were performed on the simulator. This Chapter is divided into two sections. The first one analyses the performance of the partitioning algorithm and the impact of various parameters on the scheme performance. The second section compares the performance of the scheme with the performance of other bandwidth allocation concepts, analysing the network and traffic environments that are optimal for the implementation of the scheme.

Chapter 7 provides the conclusion of the Thesis. It gives a detailed discussion of the experimental results and analyses the future research issues, especially concerning the end-to-end implementation of the Dynamic Bandwidth Partitioning concept.

CHAPTER 2

QUALITY OF SERVICE IN INTERNET

2.1. Introduction

The global telecommunication network of the future is clearly going to be based on end-to-end packet service using the Internet Protocol (IP). The advantages of connectionless design, flexibility and robustness of IP are numerous. However, problems that IP brings are not small and a lot of work is yet to be done to improve the original IP connectionless design to build a powerful and robust network for tomorrow's communication services.

New, sophisticated Internet applications, which require strict end-to-end performance guarantees, are emerging. These new applications include interactive TV on demand, distance learning, video conferencing, etc. The Internet is changing from a data network to the broadband integrated network that must be capable of carrying various types of traffic with very different requirements. The Internet has come to the point where simple connectionless IP forwarding is not sufficient. The users of the new generation of Internet applications require quality of service guarantees – they require enough bandwidth for their video streams, guaranteed small end-to-end delay for their telephone conversations, and guaranteed low loss and high throughput for their file transfers. There is a very complex problem of optimisation of the network control to satisfy diverse requirements, including the fairness for the data traffic and the performance guarantees for the real-time traffic.

This Chapter analyses the QoS problem in the Internet. Firstly, in section 2.2 we comment on the issue of quality of service itself, on what it means from the users' perspective and what from the providers' perspective. The measurement and evaluation of the network performance are also discussed in this section. Section 2.3 explains in detail the Internet architecture, which is based on the TCP/IP protocol suite. Internet Protocol, which lies in the heart of the TCP/IP network, provides only connectionless packet delivery, without precise QoS support. However, IP does have some support for QoS, which is analysed in detail in section 2.4.

The Internet community, organised in the Internet Engineering Task Force [IETF], is trying hard to solve the Quality of Service problem in IP networks. Several recent architectural propositions will be explained in section 2.4, including the Integrated Services Architecture, the Differentiated Services Architecture, and Multiprotocol Label Switching.

2.2. Quality of Service

2.2.1. Users' Perspective

What does Quality of Service in Internet mean to the end-users? In the business world, it determines whether a normal voice conversation is possible, whether a videoconference is of sufficient quality, or if a multimedia application improves productivity for the staff. At home, it determines whether the savings offered by an inexpensive voice service are worthwhile, or if there are complaints about the quality of a video-on-demand movie. Around the world, discriminating businesses and residential users demand higher quality than the best effort service offered by the initial Internet design.

It is very hard to precisely define Quality of Service. The Quality of Service of a network refers to the properties of the network that directly contribute to the degree of satisfaction that users perceive. The perceived QoS depends on the type of application the user is running. There are many examples for this: the same network can have a poor quality if you want to hear an audio signal on the Web, but can be sufficient to download a text file relatively quickly. Typical thresholds for acceptable QoS for major Internet applications are listed in Table 2.1.

Table 2.1: Thresholds for acceptable QoS [BOU00]

	Application	**Minimum user requirement**
Inelastic Applications	Video	5 frames per second
	Audio	< 30% packet loss Latency < 400ms
	Interactive real-time multimedia	Delay<200ms Jitter<200ms
Elastic Applications	Web-page access	Latency<11 seconds

If we consider video traffic, when video display devices play back frames at rates of between 25 and 30 frames per second using the image persistence provided by display systems like the cathode ray tube, the human eye-brain perceives continuous motion. When loss or errors disrupt a few frames in succession, the human eye-brain detects a discontinuity. On the other hand, looking at the perception of an audio signal, the human ear-brain combination is less sensitive to short dropouts in received speech, being able to accept loss rates ranging from 0.5 percent to 10 percent depending upon the type of voice coding employed. This level of loss may cause an infrequent clicking noise, or a loss of a syllable. The perception of a delay also depends on the type of application. One-way communication like video broadcast (television) or an audio signal (radio) can accept relatively long absolute delays. However, delay impedes two-way, interactive communication if the round-trip latency exceeds 300ms [BOU00].

Higher level protocols can also have an influence to the end-user's perception of the network performance. Many audio and video protocols tolerate errors in the received information to a certain degree, but they are highly sensitive to delay variation. Streaming audio and video protocols employ a limited playback buffer to account for delay variation. If too little or too much data arrives while the application is playing back the audio or video, then the application either starves for data or overflows the playback buffer. Some data protocols (e.g. TCP) respond to delay and loss through retransmission strategies to provide guaranteed delivery. Since many packet-switched data networks exhibit significant delay variation (especially over congested portions), applications usually insert a substantial delay before starting playback.

Selecting precise estimates for QoS parameters like loss, delay and delay variation is not an easy task. Part of the difficulty arises from the subjective nature of perceived quality. A commonly employed approach groups applications with similar QoS requirements into broad generic classes and then specifies the QoS parameters for these classes. The idea is to enable the network to guarantee a specified QoS for the traffic that conforms to a precisely defined set of parameters. End-users define this set of parameters in a traffic contract with a service provider. Such a paradigm was introduced in ATM networks, and in recent times it is known under the name Service Level Agreement (SLA) [VER99]. SLA is a service contract between customer and a service provider that specifies the forwarding service a customer should receive. SLA defines basic traffic parameters such as mean bit rate, peak bit rate, maximum burst size, maximum packet loss, etc. It can be static or dynamic. A Static SLA is negotiated in a regular basis, e.g. monthly and yearly. A Dynamic SLA uses a signalling protocol, usually RSVP [RFC2205], to negotiate the service on demand. Service Level Agreements present very interesting issue which has received large attention recently [STE01]. However, in this Thesis the problem of SLAs is only mentioned. Further analysis of the SLAs is out of the scope of the Thesis.

2.2.2. Providers' Perspective

Providers need to design a flexible network that is capable of responding to users that will have very different performance requirements. In the multiservice environment providers face a number of different applications which require different levels of network performance. Table 2.2 presents an overview of typical applications and their relative bandwidth and delay requirements on the network.

Table 2.2. Bandwidth and Delay Requirements of typical Internet Applications

	Voice	**File Transfer**	**Video Conference**	**Video Broadcast**
Average bandwidth	Very low	High	Very High	Very High
Peak bandwidth	Low	High	Very High	Very High
Delay	Very High	Low	Very High	High
Delay variation	Very High	Low	Very High	Low

We can see in table 2.2 that requirements are very different, going from 'very low' to 'very high'. Providers need to accommodate a large spectrum of different applications, and to provide an adaptive multi-service environment in which it is possible to optimise the network to provide maximum performance for all users. This is in no way a simple task, since the number of technical and business requirements that need to be answered is large.

Table 2.3 [McD00] shows the individual influence some design parameters have on the QoS parameters, such as delay, delay variation, loss and random errors. Technical issues that need to be resolved include admission control, congestion control, packet scheduling, routing and bandwidth allocation. The influence of each of these issues on the end-to-end quality of service is significant. Furthermore, all of the technical issues should be closely combined with the future concept of charging for the use of the Internet.

Table 2.3. Impact of the Design Issues on the QoS Parameters

Feature	**Delay**	**Delay Variation**	**Loss**	***Random Errors***
Propagation Delay	✓			
Router Queue Architecture	✓	✓	✓	
Link Rate	✓	✓		
Packet Size	✓	✓	✓	
Buffer Capacity	✓	✓	✓	
Resource Allocation	✓	✓	✓	
Variations in traffic load	✓	✓	✓	
Router and link failures			✓	
Bit and burst errors			✓	✓
Number of routers traversed	✓	✓	✓	✓

In summary, the problem network designers face is very complex. The network must be designed to answer a number of problems, at the same way maximising the number of users, and the revenue.

2.2.3. Utility and Utilisation

The network design objective is to build the network that can provide a large number of users the performance they can be satisfied with. In order to define precisely the network design objectives, we need to be able to evaluate the level of the network performance.

There are many ways in which we can measure the performance of the network. We want a single measure that depends on the network in question, on the profile of the offered traffic and on the quality of service requirements the offered traffic puts on the network. The most important performance measures include network utilisation, network capacity, throughput, delay, delay variation (jitter), loss probability, blocking probability for traffic flows, etc. The open question is to determine how these performance parameters influence the level of satisfaction for each individual user in the Internet.

The Internet should be designed to meet the needs of the users, and so any evaluative criteria must reduce, in essence, to the following question*: how happy does the network make the users*? The network performance should be evaluated in terms of the degree to which the network satisfies the service requirements of each user's applications. For instance, if a particular application is more sensitive about throughput than delay, or vice-versa, the network performance to that application should be evaluated accordingly.

Following this concept, it can be said that each user of an Internet application receives a certain utility (quality of service level) from the network application he is using. The user derives the utility from that application's performance in the network (e.g. the picture quality for video, the sound quality for audio application, etc.). The application's performance, in turn, depends on the nature of the network performance the application receives. We can formalise a notion of the network performance based on utility as follows. Let the vector s_i describe the service delivered to the i^{th} application or user. s_i contains all relevant measures (delay, throughput, packet drops, etc) of the delivered service. We then let the *utility function* U_i map the service delivered into the performance of the application. Increased U_i reflects improved application performance. The utility function describes how the performance of an application depends on the delivered service. The goal of network design is to maximise the performance of the resident applications. With this formalism, this goal can be restated as being, quite simply, to maximise the sum of the utilities, $\sum_i U_i(s_i)$.

In this Thesis, we are interested in the optimal bandwidth allocation in the multi-class Internet. Here the 'optimal' means the bandwidth allocation that generates maximal utility to the end-users. Similar to [GAR00], we assume that bandwidth is the single resource requirement, and analyse the perceived utility by using utility functions which map the amount of allocated bandwidth to the end-users' satisfaction with the network performance. We argue that, in the environment of differentiated services, the network *utilisation* cannot be only measured in terms of the number of traffic flows that entered the network. The optimal resource allocation scheme for the multi-class Internet is the one that generates maximal end-users' utility.

The notion of utility is crucial for the work presented in this Thesis. Before defining the scheme in Chapter 4, we will analyse the utility approach to fairness in section 3.2.2. In section 4.2 the resource allocation based on utility will be analysed. A substantial amount of work has been done on the utility-based optimisation of bandwidth allocation in both single-service and multi-service network. An overview of this work will be given in section 4.2.

At this stage, it is important to underline that the usage of the end-users' utility is not a novel approach in the literature. Lippman and Stidham [LIP77] introduced the waiting cost per unit time for each accepted customer in the system. The longer is the time customer spends in the system, the cost is higher. Kirkby et al. [KIR99] used policy based utility functions for ingress control of all types of traffic in order to ensure that only the approved volume of traffic is on the network at any one time. Kelly et al [KEL97] defined the utility function $U_r(x_r)$ of the user r that has rate x_r allocated, as an increasing, strictly concave and continuously differentiable function of x_r. Shenker and Breslau [SHE98] used utility functions to compare best-effort and reservation-capable networks.

In summary, the new multiservice Internet environment is changing our views about the network design. The objectives have changed from simple forwarding of data from the source to the destination, to sophisticated analysis of the admission, scheduling and routing controls, which are all being optimised to produce the maximal user satisfaction. Throughout this Thesis the new notion of network performance evaluation presented here will be discussed in more detail.

2.3. Internet Architecture

2.3.1. Introduction

Today's Internet is based on the TCP/IP protocol suite. Packets of Internet data called IP datagrams are carried through the network under the rules of Internet Protocol (IP). On top of IP lie transport layer protocols TCP (Transmission Control Protocol) or UDP (User Datagram Protocol). TCP is responsible for providing reliable delivery of data. Table 2.4 [BLA95] shows the TCP/IP protocol stack with roles of particular layers in the stack.

Table 2.4. Layers in TCP/IP protocol stack

Layer	Role
Layer 4: Applications Service Layer	Supports the direct interfaces to an end user application. Contains several widely used protocols, like FTP, HTTP etc.
Layer 3: Service Provider Protocol	Responsible for end-to-end communications. Contains TCP and UDP.
Layer 2: Internetworking	Provides the functions necessary for connecting networks and gateways into one coherent system. Contains IP and supporting protocols.
Layer 1: Subnetworks	Allows data to be delivered within the network. Can be Ethernet LAN, ATM, optical network…

The Internet started as a 'data only' network, a communication network which offered transport of data between end-users. The network was designed to be reliable, efficient and to ensure that the application requirements are met. The only application was file transfer, and the requirements were simple: to provide minimum possible packet loss with no regard to the end-to-end transfer duration. Thus variable and unlimited delays were acceptable. However, the future Internet is likely to be a more complicated network, with a wide variety of applications requiring very different performance from the network.

This section will describe the main features of the two protocols that form the basis for the Internet, the Internet Protocol (IP) and the Transmission Protocol (TCP). IP is presented in more detail, with special attention to the new version of IP, IP version 6.

2.3.2. Internet Protocol – IP

The Internet Protocol (IP) is the central part of the Internet protocol suite. IP has a task of routing blocks of data (IP datagrams) between the end systems that want to communicate. An IP datagram is a sequence of fields containing a header and a payload. It is interesting to note that each IP packet contains complete addressing information, and therefore each IP datagram is independently routed through the network. This is a key feature of the *connectionless* IP network, where packets belonging to the same traffic flow (connection) are simply thrown in the network, and then collected at the end-point where it is possible to put them in a correct sequence. The connectionless feature of the network makes IP network different from *connection-oriented* networks, such as PSTN or ATM networks.

2.3.2.1. IP Datagram

The current IP is known as IP version 4, IPv4. The IPv4 datagram is variable in length. IPv4 supports 32-bit long IP addresses, which are globally unique. Routing of IP datagrams is done based on the destination address only. Figure 2.1 illustrates the structure of the IPv4 datagram.

<table>
<tr><td>Version</td><td>IP HL</td><td>Type of Service</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="5">Options (0 or more words)</td><td>Padding</td></tr>
<tr><td colspan="6">Data (0 or more words)</td></tr>
</table>

Figure 2.1. IPv4 Datagram Format

Each datagram finds its own way through the network, and the only control mechanism that exists is in the information carried in the Time To Live (TTL) field of the datagram. This field specifies how many seconds the packet can be forwarded in the network before it is declared ‘dead’. The 4-bit *Version* field specifies the IP protocol version, whether it is IPv4 or the new version, IPv6. The *Type of Service* field is very interesting for the QoS support in the IP. The possible uses of this field will be explained in detail in section 2.4.1. The *Protocol* field identifies the higher-level protocol type (TCP or User Datagram Protocol, UDP). Other fields in the IPv4 include *Header Length* and *Total Length,* which define the lengths of the header and of the datagram, respectively, and *Source Address* and *Destination Address*, which contain 32-bit IP addresses of the source and the destination of the packet.

In practice the size of the datagram is limited by the size of the single frame of the underlying layer. IP does not guarantee delivery or integrity of information. There is no acknowledgement mechanism to determine whether the packet has reached its destination or not. Therefore, IP provides connectionless delivery of data.

2.3.2.2. IP Addressing and Routing

Connectionless packet switching never establishes connections of any kind. Instead, network nodes examine the address field in every packet header and forward packets along a path toward the destination by selecting an outgoing link on a hop-by-hop basis. Typically, the nodes run a distributed routing protocol that consistently determines the next hop forwarding tables to result in optimised, loop-free, end-to-end paths. Connectionless services do not guarantee packet delivery; therefore, applications rely on higher-level protocols (e.g. TCP) to perform the end-to-end error correction/detection. Additionally, higher-level protocols must also perform flow control, since connectionless services typically operate on a best-effort basis without any notion of the bandwidth allocation.

In IPv4 the IP address space is limited to 32 bits. An address begins with a network number used for routing, followed by a local, network internal address. IP supports both multicasting and broadcasting.

There are two types of equipment at the IP level, hosts and routers. A host is any end-user computer system that connects to a network. A physical host may have several local addresses and a single network address. An IP router is a dedicated computer that attaches to two or more networks and forwards packets from one to the other, based on the network portion of the destination IP address. Routers exchange network addresses as reachability information between them using different routing protocols, like Open Shortest Path First (OSPF) or Exterior Gateway Protocol (EGP), depending on where in the network hierarchy the routers are located. IP traffic within the same network can be delivered directly from host to host, whereas IP traffic to another network always passes through one or several routers.

2.3.2.3. IP version 6

IPv6 [RFC2460][HUI98] is able to solve the address size limitation of IPv4. IPv6 uses 128-bit addresses and improves addressing functionality by introducing unicast, multicast and anycast addresses. Furthermore, other new features of IPv6 include flow labelling capabilities, and specifying the delivery priority. IPv6 is not considered to be a radical change to IPv4, but rather an evolutionary step.

IPv6 has a *Priority* field in the header. This 4-bit field enables a source to identify the desired transmit and delivery priority of each packet relative to other packets from the same source. Firstly, packets are classified as being part of traffic either for which the source is providing the congestion control, or traffic for which the source is not providing congestion control. Secondly, packets are assigned one of 8 levels of relative priority within each classification.

Furthermore, IPv6 provides the means for successful definition and maintenance of a traffic flow. The IPv6 standard defines a **flow** as a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. A flow is uniquely defined by the combination of a source address and a nonzero 24-bit flow label. All packets that are to be part of the same flow are assigned the same flow label by the source.

2.3.3. Transmission Control Protocol – TCP

IP is not designed to recover from certain problems in the network. IP discards datagrams that have exceeded the number of permissible transit hops in the network (defined in the TTL field in IP header), and also discards the datagrams due to buffer overflow. Therefore, IP does not guarantee traffic delivery. Transmission Control Protocol (TCP) is the protocol that is able to provide delivery assurance.

TCP is designed to run above IP. It takes the tasks of reliability, flow control, sequencing application session, opens and closes. TCP is a connection-oriented protocol, which refers to the fact that TCP maintains status and state information about each user data stream flowing into and out of the TCP module. It provides reliable delivery of data between host computers by establishing a connection before the applications send data [RFC793]. TCP guarantees that the data is error free and in sequence. It resides in the Transport layer (layer 4) of the OSI architecture.

Data units at the TCP layer are called *segments*. TCP passes its segments to IP, which then routes the data through the network. IP receives the incoming data packets and passes them to TCP, which analyses the TCP header to determine the application recipient and passes the data to the application in sequenced order. TCP does not support multicasting or broadcasting.

As with most protocols that provide flow control, TCP uses a form of sliding-window mechanism. The flow control mechanism used by TCP is known as a credit allocation scheme, and is fully explained in [STA98]. A congestion avoidance mechanism was added to TCP in the late eighties. This mechanism is relevant to our research and will be explained in more detail in Chapter 3.

Communication using TCP transport typically occurs in the following fashion. The application consists of two parts: a client part and a server part. The server part of the application reserves a port number, which is known to the client, and waits for the client to send messages to it. The client typically uses a dynamic port and sends an initial message (a connect request) to the server with the identity of the client's receiving port. The server now knows the port number assigned to the client and can accept the request for the connection. Data can then be exchanged.

User Datagram Protocol (UDP) is another transport-layer protocol that can operate above IP. Communication using UDP can occur in the same manner as with TCP, although the connect operation is not necessary. UDP uses the IP Layer to establish end-to-end unreliable communication between two IP machines. UDP has a message structure associated with it. It takes a block of data called the message from the sender and gives the same block of data (preserving boundaries) to the receiver. Message boundaries are preserved, but messages may be lost in the network.

The structure of the TCP and UDP headers is important from the perspective of the SLA and QoS. Information in the header fields enables network devices to determine the applications that may have generated specific packets in the network.

2.4. Internet QoS

2.4.1. QoS Support at IP Layer

The most important restriction of IP is that it provides only plain, best-effort, connectionless delivery. However, there are certain mechanisms within IP that work as a support for the delivery of the quality of service. This section discusses the ability of IP to support the QoS.

2.4.1.1. Per-class Routing

Per-class routing mechanisms provide multi-level service support based on generic service classifications rather than information regarding individual customer flows. Routers supporting per-class routing do not need to maintain state information related to individual flows. Instead, these schemes use a simple form of service type marking in conjunction with an appropriate form of queue management and scheduling.

Using the IP Type of Service (TOS) field in the IP header provides a method of marking and distinguishing between different service classes as packets traverse the network. With per-class routing, IP packets are marked with the desired QoS identifier when the packets enter the network. On all subsequent interior routers, the required action is to look up the TOS field and apply the associated action on the packet. The benefit of this method is that it can be used in large-scale networks unlike per-flow routing, where each traffic flow is treated independently.

The TOS field consists of the 3-bit precedence subfield and a 4-bit TOS subfield (see figure 2.2). The TOS subfield is set by the source system to indicate the type or quality of service that should be provided, if possible, for the datagram. In practice, routers may ignore this field. However, if a router implements a TOS capability, there are three possible ways in which the router can respond to the TOS value:

- Route Selection: A routing decision could be made on the basis of type of service. For example, any datagram requesting minimised delay should not be routed through a subnetwork that includes a satellite link.
- Subnetwork Service: For the next hop, the router can request a type of service from the subnetwork that most closely matches the requested TOS. A number of networks (e.g. ATM) support some sort of type of service.
- Queuing Discipline: A router may allow TOS and precedence to affect how queues are handled. For example, a router may give preferential treatment in queues to datagrams requesting minimised delay, or a router might attempt to avoid discarding datagrams that have requested maximised reliability.

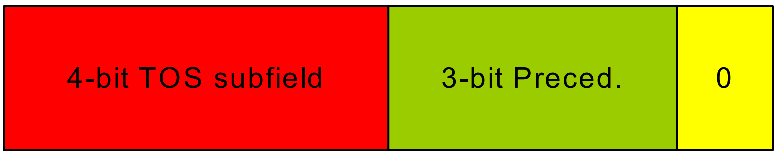

Figure 2.2. IP Type of Service Field

The Precedence subfield indicates the relative importance or priority of the datagram and is widely used by routers for the queue selection. Although the IP TOS field was intended for use by hosts and routers, it has not been as widely implemented except for IS-IS and OSPF routers [NOR00]. The use of the IP TOS field in general is currently under discussion, particularly in the light of the ongoing activity in the Differentiated Services architecture.

Important component of per-class routing is *queue management* [KLE76]. Queue management can be supported in terms of selective discarding and queuing strategies. An example of a discarding mechanism is *Random Early Detection (RED)* [FLO93], which monitors average queue length and compares it with threshold values to control the rate of random packet discards. When it comes to queuing mechanisms, *priority queuing* uses a scheduler to service certain types of traffic in preference to others. Multiple output queues are used to buffer traffic according to class. By examination of the TOS bits, a queue manager can determine the appropriate buffer an incoming packet should be stored in. An example of priority queuing is Weighted Fair Queuing [COST242][ZHA95] where traffic is classified into a number of flows that are placed into virtual queues. These are then serviced using a scheduling algorithm that takes account of packet sizes, arrival times, and current backlogs to ensure that each flow has an appropriate level of performance.

In summary, IP is able to provide means for traffic differentiation, through the use of the IP TOS field in the datagram header. Intermediate routers use the information from the TOS field and apply a certain policy on each of the traffic classes, by using different queuing management mechanisms.

2.4.1.2. QoS Routing

QoS-based routing [RAK5] is traffic routing based not only on the destination, but also on the performance requirements of the traffic flow. It does not operate as shortest-path routing currently deployed in Internet routing protocols such as OSPF [MOY98]. QoS-based routing is able to find a longer but more lightly loaded path better than the heavily loaded shorter path. It has been proven on a number of algorithms that this can be achieved [CHE98].

If the network is using shortest-path IP routing, two major problems can occur. Firstly, if a traffic flow from a source to a destination exceeds the capacity of the shortest path, the shortest path will become congested due to that single traffic flow. A second major problem is created by multiple shortest paths from different sources overlapping on a single link. If the total traffic exceeds the capacity of that link, congestion occurs.

Accurate state information is essential for efficient QoS-based routing schemes. Updating state information and calculating paths under inaccurate state information are among the most difficult and most challenging problems of QoS-based routing [GUE99]. The network state changes dynamically due to transient load fluctuation, and growing network size makes it increasingly difficult to gather up-to-date state information in a dynamic environment. If the state information is outdated, the performance of QoS-based routing can be seriously degraded and could lead to instabilities.

2.4.2. Integrated Services Architecture

The following sections present architecture models that have been developed recently. Both architectural concepts presented, Integrated Services and Differentiated Services, fully assume the multi-service nature of the future Internet, and try to define architectures that are scalable and efficient, and capable of providing the QoS guarantees to its end-users.

The Integrated Services architecture [RFC1633] relies upon a traditional datagram forwarding in the default case, but allows sources and receivers to exchange signalling messages which establish additional packet classification and forwarding state on each node along the path between them. In order to support Integrated Services there is a requirement to the routers to implement two fundamental components: *reservation of resources* and *traffic flow control*.

The Integrated Services architecture is implemented by four components: the packet scheduler, the admission control routine, the classifier and the reservation set-up protocol, usually Resource Reservation Protocol (RSVP). Integrated Services architecture is developed to support real-time as well as current best-effort services. The architecture is designed to work well with multicast as well as unicast.

The Integrated Services model proposes three service classes:

1. Guaranteed Service
2. Predictive (Controlled-Load) Service
3. Best Effort Service

Guaranteed Service [RFC2212] guarantees that datagrams will arrive within the guaranteed delivery time and will not be discarded due to queue overflows, provided the flow’s traffic stays within its specified traffic parameters. This service is intended for applications which need a firm guarantee that a datagram will arrive no later than a certain time after it was transmitted by its source. Guaranteed reservation can be created by using a reservation protocol like RSVP or by manual configuration of intermediate routers.

Predictive (Controlled-Load) Service [RFC2211] is intended to support a broad class of adaptive real-time applications, highly sensitive to overloaded conditions. These applications require reliable but not fixed delay bounds. The end-to-end behaviour provided to an application by a series of network elements that provide Controlled-Load service tightly *approximates* the behaviour visible to applications receiving best-effort service on a *lightly loaded* network from the same series of network elements.

Best Effort Service is the service of the current Internet architecture. This service doesn't provide any QoS guarantees, and QoS that end-users receive highly depends on the amount on traffic in the network. All flow-related information is present only in end-stations, and dropping packets is the only flow-related feedback from the network.

RSVP (Resource Reservation Protocol) [RFC2205] is a network control protocol that is used by the applications to require QoS for their data flows. It is a very important component of the Integrated Services architecture. RSVP runs on top of IP and relies on standard Internet routing. It is not a routing protocol, it is designed to operate with current and future routing protocols. RSVP is designed for both unicast and multicast communication in a heterogeneous network, where receivers may have different characteristics and multicast membership is dynamic. These requirements lead to a solution where the *receiver* is responsible for initiating the resource reservation.

In the RSVP protocol, the sender sends a PATH message to the receiver specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing algorithm. Upon receiving the PATH message, the receiver responds with a RESV message to request resources for the flow (see figure 2.3).

Every intermediate router along the path (the path of the RESV message is the same as the path of the PATH message) can reject or accept the request carried in the RESV message. If a request is rejected, the router will send an error message to the receiver, and the signalling process will terminate. If the request is accepted, resources will be reserved for the flow and the related flow state information will be installed in the router.

Quality of service for a particular data flow is implemented by mechanisms collectively called "traffic control". These mechanisms include Admission control, Packet classifier and Packet scheduler. *Admission control* determines whether the node has sufficient available resources to supply the requested QoS. *The Policy control* determines whether the user has administrative permission to make the reservation. If both checks succeed, parameters are set in the *Packet classifier* and in the Packet scheduler (link layer interface) to obtain the desired QoS. If either check fails, the RSVP program returns an error notification to the application process that originated the request.

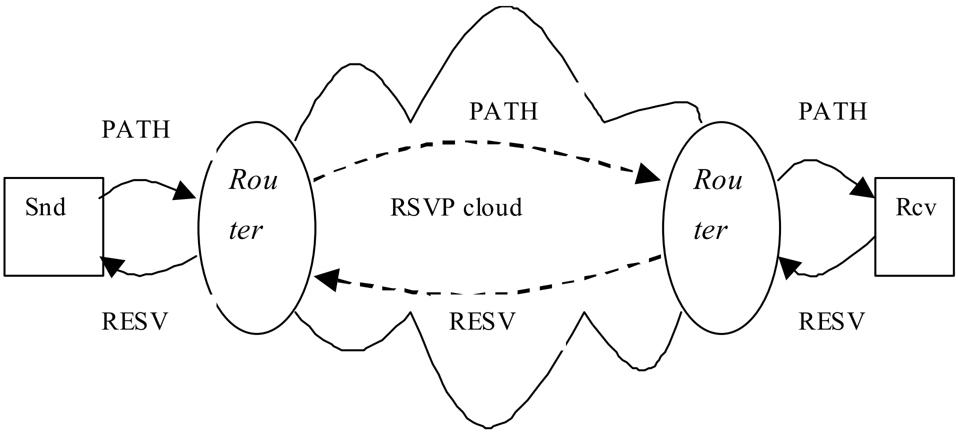

Figure 2.3. RSVP Signalling

The Integrated Services architecture is a very important step towards providing QoS guarantees in the Internet. The major strengths of the Integrated Services/RSVP approach include:

ADVANTAGES OF INTSERV/RSVP

- ***Assured QoS***: If the RSVP reservation is successful, a connection can obtain an assured level of service from the network
- ***Automatic adjustment to route changes****:* Because RSVP messages follow the same route as normal datagrams in the network, they are able to reserve capacity along the right path without making assumptions regarding how the routing protocols work.

However, this architecture has a number of disadvantages. The most important disadvantages are:

DISADVANTAGES OF INTSERV/RSVP

- ***Scalability****:* In the absence of state aggregation, the amount of state information in each node scales in proportion to the number of concurrent reservations, which can be potentially large on high-speed links.
- ***Requirement on router is high:*** Each router must implement RSVP, admission control and packet scheduling.

- ***Reservation Latency:*** The sender is only assured of reservation on the receipt of the first RESV message from the downstream router. This implies that the approach would not work too well for short-lived connections.
- ***Lack of Universality:*** End-to-end QoS can only be guaranteed if all the routers and hosts along the routed path are running the RSVP software, because tunnelling through non-RSVP clouds destroys the end-to-end QoS.

2.4.3. Differentiated Services Architecture

The Differentiated Services (Diff-serv) architecture [RFC2475] offers a framework within which providers can offer each customer a service differentiation. A 'service' is defined as the overall treatment of a defined subset of customer traffic within the network.

In the Diff-serv (DS) architecture (figure 2.4), the network is consisted of DS domains, contiguous sets of nodes which operate with a common set of service provisioning policies. Customers request a specific performance level on a packet by packet basis, by marking the DS field of each packet. The DS field is a TOS field of the IPv4 header or a Traffic class field of the IPv6 header [HUI98]. The first 6 bits of the DS field form a DS codepoint. Packets are classified in one of a small number of aggregated flows, based on the setting of bits in the DS codepoint. An aggregated flow is a number of flows that share forwarding state and a single resource reservation along a sequence of routers.

A DS codepoint value specifies the Per-hop behaviour (PHB) that packets receive on nodes along their path in the DS domain. PHBs are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams. Therefore, QoS in the Diff-serv architecture can be deployed *without* any end-to-end signalling.

The Differentiated Services standards propose two different Per-hop Behaviours:

- Expedited Forwarding PHB
- Assured Forwarding PHB

The definition of PHBs is a critical part of the work in defining the Differentiated Services architecture. The *Expedited Forwarding* (EF) PHB [RFC2598] can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. The EF traffic should receive this rate independent of the intensity of any other traffic attempting to transit the node.

Assured Forwarding (AF) PHB [RFC2597]: This PHB group provides delivery of IP packets in four independently forwarded AF classes. Within each class an IP packet can be assigned one of three different levels of drop precedence. In case of congestion, the drop precedence of a packet determines the relative importance of the packet within the AF class. A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with higher drop precedence.

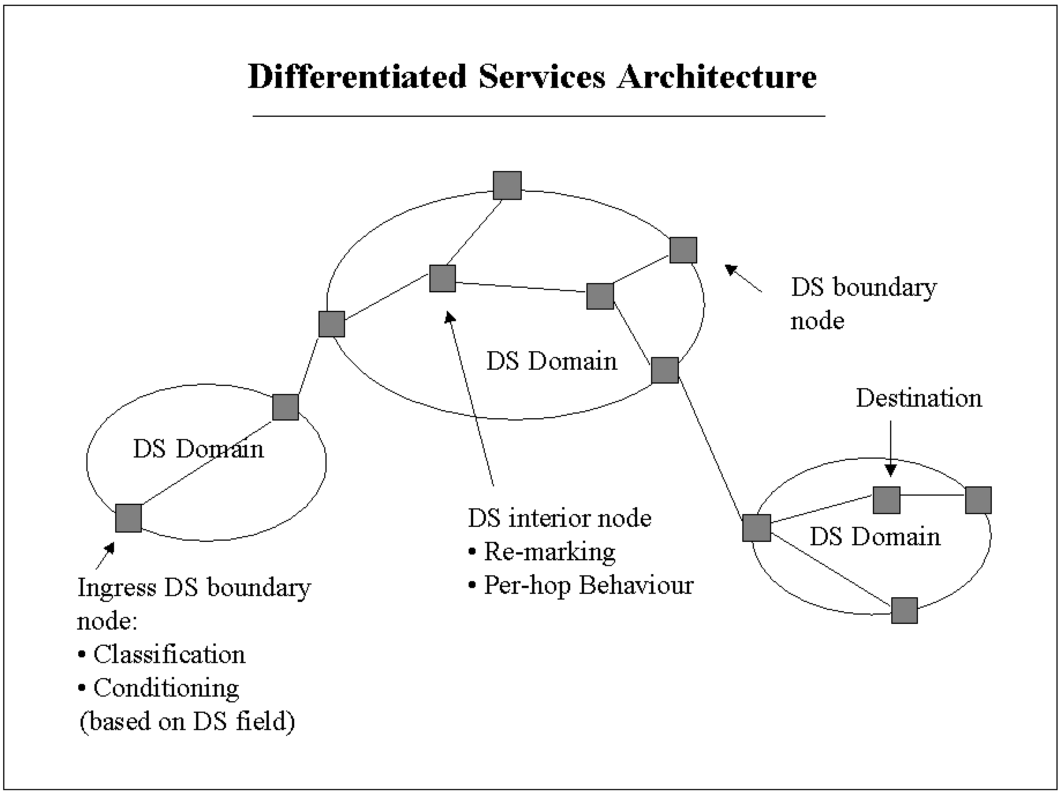

Figure 2.4. The Differentiated Services Architecture

At the DS boundary nodes (DS nodes that connect a DS domain to a node that is in another DS domain, or in a domain that is not DS-capable), the *classification* and *conditioning* of the incoming traffic is done. The packet classification policy identifies the subset of traffic which may receive a differentiated service by being conditioned and/or mapped to one or more behaviour aggregates (by DS codepoint remarking) within the DS domain. Traffic conditioning performs metering, shaping, policing and/or remarking to ensure that the traffic entering the DS domain conforms to the rules specified.

Differentiated Services is significantly different from Integrated Services. Since service is allocated in the granularity of a class, the amount of state information is proportional to the number of classes, not proportional to the number of flows. Differentiated Services therefore provides a scalable QoS solution to ISP networks. Furthermore, sophisticated classification, authentication, marking, policing and shaping operations are only needed at boundary. ISP interior routers need only to implement classification and some simple scheduling algorithm. In the DiffServ architecture, since there is no signalling or per-flow control, performance guarantees rely on accurate dimensioning, and the use of policers at the edge of the network to ensure that users remain with their agreed profiles. Therefore, Differentiated Services is easier to implement and deploy than Integrated Services.

The end-to-end solution in the DiffServ approach is based on top-down provisioning, with centralised intelligence (called the *Bandwidth Broker*) collecting state information and configuring each network device.

The Differentiated Services is a powerful and scalable solution for traffic handling. It recognises that many applications do not require the high-quality end-to-end performance guarantees. Instead, proper network provisioning and coarse traffic classification into a small number of "priority" classes is sufficient as the applications can adapt to changes in network conditions. The main premise behind this is that only a small portion of the applications require strict service guarantees. With proper network provisioning, and protection from other lower priority traffic, these applications would get their desired service. The benefits are higher network utilisation as more flows can get through, and greater simplicity as only minimal signalling and simple data forwarding mechanisms are required. On the other hand, the low overhead of Diffserv approach, especially compared with the per-flow configuration in IntServ architecture outweighs the compromise in performance guarantees. That is why DiffServ is considered to be a much better end-to-end scalable solution for the integrated network.

However, there are problems with the DiffServ solution. In a comprehensive analysis, Gibbens et al [GIB00] drew the conclusion that '*DiffServ QoS mechanism is unlikely to be able to provide real measurable distinctions between classes on a pure IP network which has no access restrictions, without either bandwidth partitioning at a lower layer (or through a facility like MPLS), or gratuitously damaging some traffic'*. It is becoming clear that, for an efficient end-to-end solution, Differentiated Services architecture requires some additional mechanisms, particularly for providing strict QoS guarantees to applications that require such guarantees.

2.4.4. Multiprotocol Label Switching (MPLS)

MPLS was originally designed as a way of improving the forwarding speed of routers but is now emerging as a crucial standard technology that offers new capabilities for large-scale IP networks. The essence of MPLS [RFC3031] is the generation of a short fixed-length *label* that acts as a shorthand representation of an IP packet's header. A label is a relatively short, fixed-length, unstructured identifier that is inserted in front of each data packet on entry into the

MPLS network. A labelled packet does not carry a single label. It carries a number of labels, organised as a LIFO (last-in-first-out) stack. This stack is called the *label stack* [RFC3032]. In MPLS every forwarding decision is based exclusively on the label at the top of the stack. The existence of the label stack is very useful in creating explicit routes. The label stack is represented as a sequence of label stack entries. Each label stack entry is represented by 4 octets (see figure 2.5).

The label stack entries appear after the data link layer headers, but before any network layer headers. The top of the label stack appears earliest in the packet, and the bottom appears latest. The network layer packet immediately follows the label stack entry which has the S bit set.

MPLS routers are called *Label switch routers* (LSRs). LSRs use link-level forwarding to provide a simple and fast packet-forwarding capability. Label-swapping packet forwarding is based on a simple short-label exact match. The most important thing about label-based forwarding is that only a single forwarding algorithm is needed for all types of switching and this can be implemented in hardware for extra speed.

LABEL	EXP	S	TTL

Label value, 20 bits

Exp, Experimental, 3 bits

S: Bottom of stack, 1 bit

TTL: Time to live, 8 bits

Figure 2.5. MPLS Label stack entry, 32 bits

The path that data follows through the network is called a *Label switched path* (LSP). To enable the use of LSPs, the forwarding tables at each LSR must be populated with the mappings from {incoming interface, label value} to {outgoing interface, label value}. This process is called LSP set-up, or *Label Distribution*. In general, an MPLS node receives an “outgoing” label mapping from the peer that is next hop for a stream, and allocates and distributes “incoming” labels to upstream peers for a given stream.

At the ingress to an MPLS network, the *MPLS edge router* examines each packet to determine which LSP it should use and what label to assign to it. At all the subsequent nodes within the network the MPLS label, and not the IP header, is used to make the forwarding decision for the packet. The label decision is a local matter but is likely to be based on factors including the destination address, the QoS requirements and the current state of the network. Labels are normally local to a single data link and have no global significance (as would an address). For example, a label could correspond to an ATM VPI/VCI [ONV94], or a DWDM wavelength for optical networking.

When a core LSR receives a labelled packet, the label is first extracted and it is used as an index into the forwarding table that resides in the LSR. When the entry indexed by the incoming label is found, the outgoing label is extracted and added to the packet and the packet is then sent out the outgoing interface(s) to the next hop(s) that are specified in the entry (multicast involves multiple outgoing packets). All subsequent nodes in the network use the label for their forwarding decisions. The value of label may, and usually does, change at each LSR in the path through the network.

The MPLS architecture does not define a single protocol for the distribution of labels between LSRs. The underlying principles in label distribution are that an LSP is set up either in response to a request from the ingress LSR (downstream-on-demand), or pre-emptively by LSRs in the network, including the egress LSR (downstream unsolicited).

There are currently two label distribution protocols that provide the necessary support: Resource Reservation Protocol (RSVP, [RFC2205]), and Constraint-based Routed Label Distribution Protocol (CR-LDP, [JAM01]). These two protocols provide the similar level of service, but the way they operate is different. CR-LDP is a set of extensions to Label Distribution Protocol (LDP, [RFC3036]) specifically designed to facilitate constraint-based routing of LSPs. Constraint-based Routing computes routes that are subject to constraints such as bandwidth and administrative policy. Like LDP, CR-LDP uses TCP sessions between LSR peers and sends label distribution messages along the sessions. RSVP uses a message exchange (messages are IP datagrams) to reserve resources across the network. The extensions to RSVP for LSP Tunnels [AWD01] enhances generic RSVP so that it can be used to distribute MPLS labels. The key differences between CR-LDP and RSVP are the reliability of the underlying transport protocol and whether the resource reservations are done in the forward or reverse direction.

The rapid speed of developments in both IP networks and optical technologies are inevitably bringing the two domains closer together. The benefits of optical WDM networks include larger bandwidth, better network scalability, and more efficient operations. It has become very desirable to design an integration strategy that will efficiently combine the layer 2 and layer 3 capabilities of IP/MPLS networks and huge benefits of the WDM-based optical systems.

MPLS is particularly interesting because it works as a powerful tool for traffic engineering. Traffic engineering can be defined as an iterative process of network planning and network optimisation. The purpose of traffic engineering is to optimise resource efficiency and network performance. It refers to the process of selecting the paths chosen by data traffic in order to balance the traffic load on the various links, routers and switches in the network. QoS schemes presented in this Thesis, Integrated Services and Differentiated Services, provide degradation of performance when traffic load is heavy. Therefore, minimising congestion and rerouting traffic from the heavy-load parts of the network is a major traffic and resource oriented performance objective. The interest here is not on transient congestion resulting from instantaneous bursts, but rather on congestion problems that are more prolonged.

MPLS and *constraint-based routing* have proved to be powerful tools in traffic engineering. The use of constraint-based routing and explicit routing is the key of traffic engineering in MPLS. Precise definition of performance requirement, efficient route calculation, and setting the bandwidth requirements on every LSP gives enough working space for efficient traffic engineering in MPLS.

2.5. Summary

This Chapter presented the review of the quality of service problem in the Internet. The Internet has been originally designed to be a data-only network. Connectionless forwarding of the Internet Protocol (IP) provided sufficient efficiency for such a network, since the end-users of a data network do not have strict requirements concerning the delay and delay variations bounds.

However, the current Internet needs to evolve to a sophisticated broadband communication network which is capable to transport various traffic types and to provide service to numerous applications. Sophisticated applications, namely video telephony, video-on-demand, etc. require strict performance guarantees. The problem of Quality of Service became visible and very important.

This Chapter discussed the quality of service issue, both from the users' and providers' perspective. It also reviewed the Internet architecture, most of all the major features of IP and TCP protocols.

Section 2.4 discussed the current ideas and standards for the QoS-aware Internet, The limited QoS support provided on the IP level is firstly analysed. This is followed with detailed explanation of several Internet architectures and mechanisms, namely Integrated Services with RSVP, Differentiated Services and Multiprotocol Label Switching. Although these architectures are not overwhelmingly accepted in the Internet community, and although none of them will serve as the *unique* solution to the quality of service problem, they are still very important, because they present a foundation for the further research that is being done in the area, including the research presented in this Thesis.

CHAPTER 3

BANDWIDTH ALLOCATION IN THE MULTI-CLASS INTERNET

3.1. Introduction

This Chapter analyses bandwidth allocation for both single-service and multi-service Internet. Bandwidth allocation for the single-service network is based on the notion of fairness. In contrast, the main issue behind the bandwidth allocation problem in the multi-class network is the integration of different classes of traffic. Each traffic class has different requirements from the network, and the network must be designed to allocate its bandwidth between traffic flows from all traffic classes in a way to accommodate all the requirements while allowing high network utilisation.

The study presented in this Chapter serves as introduction to the new bandwidth allocation scheme, Dynamic Bandwidth Partitioning, which will be presented in the next Chapter.

The way bandwidth is shared among the users of the communication network determines the performance level the network will be able to achieve. If the network resources are shared in an inefficient way, a number of users will be dissatisfied with the network performance, while a large amount of resources will stay under-utilised.

IP networks were primarily designed to deal with data communications. The connectionless delivery provided by IP does not provide any performance guarantees to the end-users. What current Internet offers is a single class of ‘best-effort’ service. All users just send their data in the network, the network resources are shared ‘fairly’, where each traffic flow receives approximately equal share of the bottleneck capacity. The network does not have any admission control.

On the other hand, the number of new real-time Internet applications is increasing rapidly, and the possible number of users for these applications is huge. These applications have strict performance (QoS) requirements from the network. If the network is unable to guarantee them the performance they require, not only that the users will not be satisfied, but also they will refuse to use the network in the future. Considering the high revenues that can be made from the sophisticated Internet applications of the future (such as playing distributed computer games, TV on demand and video telephony) it is quite obvious that no network operators can afford that to happen. The Internet needs additional, architectural improvements, which can bring sophisticated real-time applications to the end-users with the guaranteed level of quality of service.

The general objective in bandwidth sharing is to use all available bandwidth while trying to achieve fairness. Different approaches to measure the fairness are analysed in section 3.2. The special focus of that section is on the utility approach to fairness. This is very important for our research since in our Dynamic Bandwidth Partitioning scheme the decisions about the change in the bandwidth partitioning are made based on the measured level of user-utility.

Bandwidth allocation in the Internet is actually done in the network by using different bandwidth sharing algorithms. There are broadly two classes of adaptive bandwidth sharing algorithms:

- Explicit rate calculation
- Congestion indication.

Explicit rate calculation assumes a centralised controller that computes, for example, max-min fair shares for all routes. Explicit rate calculation done by a centralised controller is, however, impractical in the real networks. Practical explicit rate algorithms are based on the distributed calculation of rate algorithms. More about those algorithms can be found in [ROB99].

Most network flow control protocols are based on simple binary indications of congestion issued independently by the network links. For example, in TCP congestion control users increase their sending rate linearly until congestion occurs and then begin to decrease the rate exponentially.

Section 3.3 analyses congestion control in more detail. After the overview of the congestion control, the use of linear control algorithms in the end-to-end congestion control is presented. This is relevant to our research since the dynamic change of partitioning parameters in the Dynamic Bandwidth Partitioning scheme follows a simple linear control rule. Finally, section 3.3.3 sets out the analysis of TCP congestion control, which is important for us because of the implementation issues for the Dynamic Bandwidth Partitioning.

Section 3.4 gives an overview of bandwidth allocation schemes in the Internet. Two basic concepts - complete sharing and complete partitioning, are analysed, with different approaches for the performance enhancement of them. The analysis presented in section 3.4 puts our work in the broader framework, by presenting many different schemes that have been proposed. Later chapters of the thesis will make many references to this section, since we are using some of the approaches presented there for the efficiency evaluation of Dynamic Bandwidth Partitioning.

3.2. Fairness

This section discusses different approaches to fairness. The objective of resource management is generally to use the available resource while trying to achieve fairness in the way the resource is shared between the users. We study a particular case of bandwidth allocation in the communication networks here. There are different forms of fairness, and the important ones will be analysed in this section.

3.2.1. Max-Min fairness

The most common form of fairness is max-min fairness. In a max-min fair system all connections get the same share of a bottleneck. If a connection cannot use its entire share, e.g. because it has a slower rate in another bottleneck, then the excess capacity is shared fairly among the other connections.

Let us consider a network as a set of links L where each link l has a capacity C_l. A number of flows compete for the access to the network. We note ξ_r as the rate of flow $r \in R$, where R is the set of flows, where each flow is defined by its route in the network. The bandwidth sharing objective can be defined as finding the set of ξ_r to meet some defined sharing objective. In max-min fairness, rates are made as equal as possible subject only to the constraints imposed by link capacities [NAH98]. In other words, each flow uses the bandwidth that is allocated to it on its bottleneck link. More formally, for every route r, there is at least one link $l \in r$ such that

$$\sum_{r_l \ni l} \xi_{r_l} = C_l \qquad (3.1)$$

Link l then represents the bottleneck for all the flows r_l. The capacity of a flow r is then the minimal bottleneck capacity on its route:

$$\xi_r = \min\{\xi_{r_l}\} \qquad (3.2)$$

3.2.2. Utility Approach to Fairness

Max-min fairness maximises the bandwidth allocated to each traffic flow. However, there appears to be no clear economic reason why max-min sharing should be preferred. A more rational objective would be to maximise overall end-user's utility or to minimise the expected response time of any transfer. These objectives are not necessary met by using max-min fairness [ROB99]. Another problem with max-min fairness is that, while max-min fairness is often the stated objective, it is widely recognised [ROB99] that this is imperfectly achieved by most network flow control protocols. This is very important since the bandwidth sharing mechanism should be not only efficient and fair, but simple, too.

Proportional fairness is an example of a more general fairness concept, called the "utility" approach. The notion of utility has already been introduced in the section 2.2.3. Every source s has a utility function u_s where $u_s(x_s)$ indicates the value to source s of having the rate x_s. Every link l (or network resource in general) has a cost function g_l, where $g_l(f_l)$ indicates the cost to the network of supporting an amount of flow f_l on the link, where $f_l = \sum_{s=1}^{S} A_{l,s} x_s$, and $A_{l,s} \in \{0,1\}$ is indicating whether the connection s is using the link l. Then, a "utility fair" allocation of rates is an allocation which maximises $H(\vec{x})$, defined by

$$H(\vec{x}) = \sum_{s=1}^{S} u_s(x_s) - \sum_{l=1}^{L} g_l(f_l) \qquad (3.3)$$

over the set of feasible allocations.

Proportional fairness corresponds to $u_s = \log x_s$ for all s, and $g_l(f_l) = 0$ for $f < c_l$, $g_l(f_l) = +\infty$ for $f_l \geq c_l$. Rate proportional fairness corresponds to $u_s(x_s) = \omega_s \, ln(x_s)$, and the same choice of g_l. Computing utility fairness requires solving constrained optimisation problems.

Very important work on utility approach to fairness has been done by Cao and Zegura [CAO99]. They introduced *utility max-min* fairness. Informally, a feasible bandwidth allocation vector X is *utility max-min* fair if, for each connection i, its utility $u_i(x_i)$ cannot be increased while maintaining feasibility, without decreasing the utility $u_j(x_j)$ for some session j which satisfies $u_j(x_j) \leq u_i(x_i)$. It is interesting to note that the utility max-min fairness is equivalent to max-min fairness if the utility functions for all applications are the same.

The utility approach to fairness is very interesting for our Dynamic Bandwidth Partitioning scheme. Work of Cao and Zegura will be explained in more detail in section 4.2 in the next Chapter, along with other research done on utility as the performance measurement for communication networks.

3.2.3. Proportional Fairness

Kelly et al [KEL97] claim that bandwidth allocation schemes should maximise the overall utility of rate allocations assuming each route (each user) has a logarithmic utility function. This optimisation results in proportional fairness. A system is proportionally fair if the aggregate of proportional rate ξ_r changes with respect to any other feasible allocation $\xi_r^{'}$ is negative, i.e.,

$$\sum \frac{\xi_r^{'} - \xi_r}{\xi_r} \leq 0 \qquad (3.4)$$

In other words, any change in the allocation must have a negative average change. Therefore, if a source is not able to use one Nth of the bottleneck it may still be allocated less than its maximum, say 5% less, if this allows a larger, say more than 5%, increase of the rate of another connection [CRO98]. Kelly et al [KEL97] proved that rate control based on additive increase and multiplicative decrease, as in TCP, achieves proportional fairness.

Furthermore, Le Boudec proved (Theorem 1.2.3 in [LEB]) that there exists a unique proportionally fair allocation. It is obtained by maximising $J(\vec{x}) = \sum_s ln(x_s)$ over the set of feasible allocations. This conclusion is in accordance with the previous section, where it was shown that proportional fairness corresponds to utility functions $u_s(x_s) = \log x_s$.

The concept of proportional fairness can be easily extended to weighted proportional fairness [ROB99], where each traffic flow has a weight assigned to it. Kelly et al [KEL97] showed that it is possible to create weighted proportionally fair sharing using a common multiplicative decrease factor and an additive increase rate proportional to the required weight. Furthermore, it has been proved [KEL97][CRO98] that in a weighted proportionally fair system where the weights are the prices the users pay per time unit, when each user chooses the price that maximises the utility he gets from the network, the system evolves to a state where the total utility of the network is maximised. That is a typical example of local optimisations leading to a global optimum. The only constraint on that function is that the utility has to be an increasing, concave and differentiable function of the bandwidth. That is the feature of elastic traffic, but not of the real-time traffic, as will be explained in more detail in Chapter 5.

Therefore, a very interesting question is what happens in the multiservice environment, when we have different applications, with different utility functions. As we will see later in this Thesis, the real-time applications are modelled with utility functions that are not always concave. Furthermore, in the multiservice environment, admission control procedure, even if it is imposed only for one traffic class, introduces new constraints on the resource sharing mechanism, and requires a new view on the meaning of fairness.

3.3. Congestion Control

3.3.1. Overview

In a packet network, sources should limit their sending rate by taking into consideration the state of the network. Ignoring this may put the network into *congestion collapse*. Congestion collapse is a phenomenon of severe service degradation due to heavy load in the network, when buffers in intermediate routers get full and incoming packets are being discarded which causes degradation. This phenomenon did happen in the Internet in the middle of the eighties. At that time, there was no end-to-end congestion control in the TCP/IP protocol suite. The original solution for the problem was introduced through congestion avoidance mechanisms that are now required in TCP implementations. These mechanisms operate in the hosts to cause TCP connections to slow down the transmission during congestion.

In the current multi-class Internet environment, the congestion is experienced in multiple ways. The users of the applications that require guaranteed performance experience congestion by suffering from connection (call) rejections. Best-effort users experience congestion when they receive very low throughputs. The objective of congestion control is to avoid such inefficiencies.

Congestion in a network creates obvious problems for the end systems: reduced availability and throughput, and lengthened response times. Internet routing algorithms can spread the load among the routers and networks to relieve the congestion. However, these measures are only effective for dealing with unbalanced loads and brief surges in traffic. Ultimately, congestion can only be controlled by limiting the total amount of data entering the network to the amount that the network can carry. This is the underlying objective of all congestion control mechanisms.

Congestion typically manifests under two scenarios:

- When network resources are insufficient or inadequate to accommodate offered load
- When traffic streams are inefficiently mapped onto available resources, causing subsets of network resources to become over-utilised while others remain under-utilised

The first type of congestion problems can be addressed through capacity expansion and classical congestion control techniques which regulate the demand such that it fits onto available resources. The second type of congestion problems, namely those resulting from inefficient resource allocation, can usually be addressed through Traffic Engineering.

In connection-oriented networks, congestion control is either *hop-by-hop* or *rate*-based. In the rate-based congestion control, the sources know an explicit rate at which they can send. The rate may be given to the source during a negotiation phase; this is the case with ATM or RSVP. In such cases, we have a network with reservation. Alternatively, the rate may be imposed dynamically on the source by the network; this is the case for the ABR class in ATM.

3.3.2. Linear Control for End-to-end Congestion Control

Contrary to the connection-oriented networks, in the Internet congestion control is done on an *end-to-end* basis. In end-to-end congestion control a source continuously obtains feedback from all downstream nodes it uses. The feedback is piggybacked in packets returning towards the source, or it may simply be the detection of a missing packet. Sources react to negative feedback by reducing their rate, and to positive feedback by increasing it. The difference with hop-by-hop control is that the intermediate nodes take no action on the feedback; all reactions to feedback are left to the sources. The algorithms for rate increase/decrease in the end-to-end congestion control usually belong to the family of *linear control* algorithms.

Let us consider a simplified network model. Assume J sources, labelled $j = 1,...,J$ send data at a time dependant rate $x_j(t)$, into a network constituted of one link, of capacity c. We assume that time is discrete, and that the feedback cycle lasts exactly one time unit. During one time cycle, the source rates are constant, and the network generates a binary feedback signal $y(t) \in \{0,1\}$. This feedback signal is sent to all sources. Sources react to the feedback by increasing the rate if $y(t) = 0$, and decreasing if $y(t) = 1$. We further assume that the feedback carries the information on the congestion on the link, and is defined by

$$y(t) = \left[\mathit{if} \left(\sum_{j=1}^{J} x_j(t) \leq c \right) \mathit{then\ 0\ else\ 1} \right] \qquad (3.5)$$

The value c is the target rate which we wish the system not to exceed. At the same time we wish that the total traffic be as close to c as possible. The users cooperate with the system and change their rate by an amount which is a function of both the current rate, $x(t)$, and the feedback information $y(t)$. Formally,

$$x(t+1) = x(t) + f(x(t), y(t)) \qquad (3.6)$$

In general, the control function $f(t)$ can be any linear or non-linear function. However, both TCP congestion control and the control algorithm we are using in Dynamic Bandwidth Partitioning use linear control functions. That is why here we analyse only the linear functions. The equation (3.6) now transforms to

$$x(t+1)=\begin{cases} a_i x(t)+b_i, \text{when } y(t)=1 \text{ (increase)} \\ a_d x(t)+b_d, \text{when } y(t)=0 \text{ (decrease)} \end{cases} \qquad (3.7)$$

We can recognise four basic linear control mechanisms. They are presented in Table 3.1.

The objective is to make adaptation algorithm converge towards a fair allocation. In the simple case analysed here, there is one single bottleneck and all fairness criteria are equivalent. At equilibrium, we should have $x_j = \frac{c}{J}$. In their well-known paper [CHI89], Jain and Chiu analysed the fairness of linear control algorithms. We will provide only a conclusion of their analysis here.

Consider a linear adaptation algorithm of the form in equation (3.7). In order to satisfy efficiency and convergence to fairness, we must have a *multiplicative decrease* and a non-zero additive component in the increase. If we want to favour a rapid convergence towards fairness, then the increase should be *additive only*. It should be noted that the value of a_d (the multiplicative decrease factor) plays an important role. A value close to 1 ensures smaller oscillations around the target value; in contrast, a small value of a_d can react faster to decreases in the available capacity.

Table 3.1. Linear Control Mechanisms

Linear Control	Equations	Constants
Additive Increase / Additive Decrease	$x(t+1)=\begin{cases} x(t)+b_i, & y(t)=1 \\ x(t)+b_d, & y(t)=0 \end{cases}$	$a_i=1$, $a_d=1$, $b_i>0$, $b_d<0$
Additive Increase / Multiplicative Decrease	$x(t+1)=\begin{cases} x(t)+b_i, & y(t)=1 \\ a_d x(t), & y(t)=0 \end{cases}$	$a_i=1$, $a_d<0$ $b_i>0$, $b_d=0$
Multiplicative Increase / Additive Decrease	$x(t+1)=\begin{cases} a_i x(t), & y(t)=1 \\ x(t)+b_d, & y(t)=0 \end{cases}$	$a_i>1$, $a_d=1$ $b_i=0$, $b_d<1$
Multiplicative Increase / Multiplicative Decrease	$x(t+1)=\begin{cases} a_i x(t), & y(t)=1 \\ a_d x(t), & y(t)=0 \end{cases}$	$a_i>1$, $a_d<1$ $b_i=0$, $b_d=0$

This analysis of the linear control rules for rate adaptation is very important for our work. It will be shown in the next section that TCP congestion control uses the Additive Increase /

Multiplicative Decrease rule for its rate adaptation algorithm. On the other hand, the Dynamic Bandwidth Partitioning scheme employs the Additive Increase / Additive Decrease linear control for the partitioning updates.

3.3.3. Congestion Control in TCP

In the TCP/IP protocol suite, IP was kept as simple as possible, so that the network layer could focus on routing data from the source to the destination. The job of turning an exchange of the datagram connection into a solid, reliable application-to-application data connection is carried out by TCP. Among other functions, TCP performs congestion control [FEI99].

The principles of the TCP congestion control are the following [LEB][RFC2001]:

- The rate of a TCP connection is controlled by adjusting the window size
- Additive Increase, Multiplicative Decrease principle is used
- The feedback from the network is packet loss

TCP uses the *sliding window protocol concept,* in which there is a window size W (in bytes) equal to the maximum number of unacknowledged data a source may send. In TCP the receiver regulates the rate by which the sender is transmitting data, since the sender cannot transmit the next data segment if the acknowledgement of the receipt of the previously sent segment has not arrived. Let T be the average time the sender needs to wait for an acknowledgement to come back. If we assume the source uses FIFO buffer, by Little's formula [GRO85] the throughput of the TCP connection is given by [LEB]:

$$\theta = \frac{W}{T} \qquad (3.8)$$

This means that, if T is fixed, then the throughput is controlled by controlling the window size W. However, T is not a constant value in the network, it depends on the congestion status and queuing delays. Therefore, the simplest congestion control that exists in the Internet is in the fact that the sources decrease their rate when the value T increases, i.e. when the time for acknowledgements increases.

In extension to this, TCP defines a variable called congestion window, $cwnd$. The window size is given by

$$W = \min(cwnd, offeredWindow) \qquad (3.9)$$

where the $offeredWindow$ is the window size advertised by the destination. In contrast, the value for $cwnd$ is computed by the source. A TCP connection is, from a congestion point of view, in one of the three phases:

- *Slow start* – after a loss detected by timeout
- *Fast recovery* – after a loss detected by *fast retransmit*
- *Congestion avoidance* – in all other cases.

TCP congestion avoidance is interesting for us because of the use of the linear control mechanism, in this case additive increase, multiplicative decrease. Linear control mechanism is used in our Dynamic Bandwidth Partitioning scheme for the dynamic change of the partitioning parameters. In order to simplify the description of the TCP congestion control principle, which would leave more space for the analysis of the linear control algorithms, this section will describe the principles of the slow start and congestion avoidance, with only a short reference to the fast recovery/fast retransmit algorithms.

The relation between the slow start and congestion avoidance is shown in Fig. 3.1. The flow on a TCP connection should obey a principle of "*conservation of packets*" [JAC88]. This means that the new packet is not put on the network until an old packet leaves. The congestion avoidance algorithm is used only when the TCP connection obeys this principle. The slow start algorithm is used if the connection is not in the equilibrium. An additional variable called the target window [LEB], or a slow start threshold size [RFC2001], is introduced. We denote this variable as $twnd$.

In the slow start algorithm, when a new connection is established the congestion window $cwnd$ is initialised to one segment. The sender can transmit up to the minimum of $cwnd$ and the advertised window, which is imposed by the receiver. Each time an acknowledgement is received, $cwnd$ is increased by one segment. This provides exponential growth. The algorithm itself is fast enough, enabling the window to open quickly enough to have a negligible effect on the performance [JAC88]. The slow start algorithm ends when intermediate routers start discarding packets, or when the target window $twnd$ is reached.

If the target window is reached, $cwnd = twnd$, the value for both of them increases slowly, following the additive increase rule. This continues until a packet loss is detected (a timeout occurs, the acknowledgement has not been received). At that point, $twnd$ decreases rapidly, being divided by 2 (multiplicative decrease), $cwnd$ is initialised to 1, and the slow start begins again.

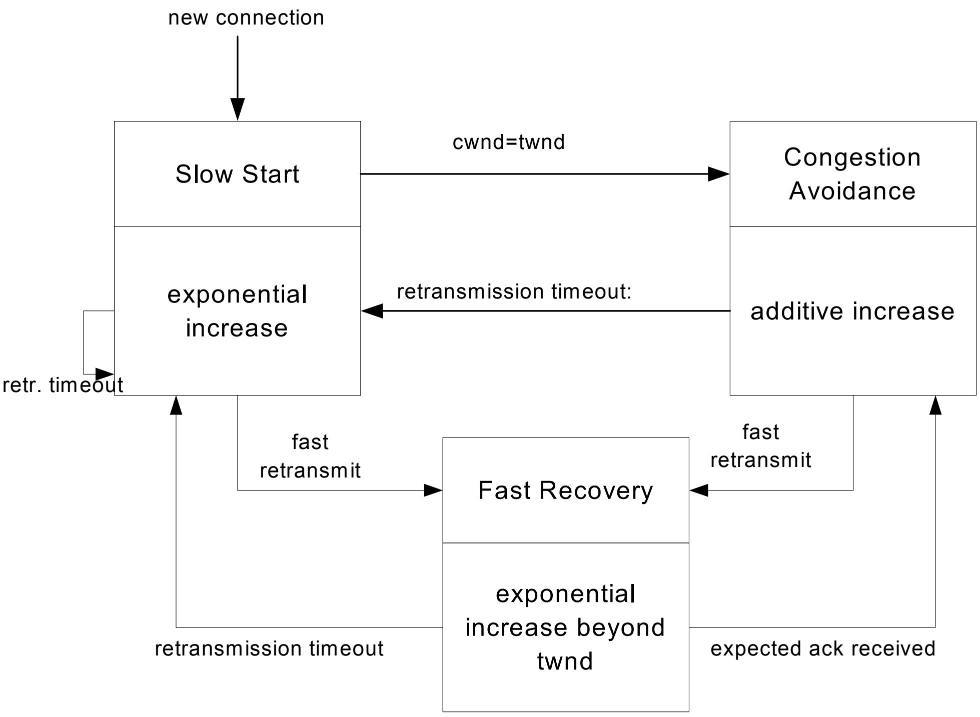

Figure 3.1. TCP Congestion Control Mechanism

The problem is that getting into the slow start phase after an *isolated* packet loss imposes a penalty, since it takes some time to reach the congestion avoidance phase again. This is why there are two specialised procedures called *Fast Recovery* and *Fast Retransmit,* which repair the isolated losses. A strong indication that a segment has been lost in the network is the receipt of three or more duplicate acknowledgements. If that happens, TCP performs a retransmission of what appears to be the missing segment (*fast retransmit).* After that, congestion avoidance, and *not* the slow start is performed. This is the concept of the *fast recovery* mechanism. The reason for not performing the slow start mechanism is that the receipt of the duplicate acknowledgements tells TCP not only that a segment has been lost but also that the following segment has been received. This shows that there is still the data flow between the two ends, meaning that the level of the congestion is still not severe enough for the slow start mechanism to be implemented.

TCP congestion control and avoidance algorithms are based on the notion that the network is a black box. The network's state of congestion is determined by the end-systems probing for the network state by gradually increasing the load on the network until the network becomes congested and the packet is lost. Active queue management (AQM) mechanisms detect congestion before the queue overflows, and provide an indication of this congestion to the end nodes. AQM mechanisms may use one of several methods for indicating congestion to end nodes. One is to use drop packets. However, AQM allows the router to separate policies of queuing or dropping packets from the policies for indicating congestion.

In summary, this section analysed congestion control in detail, presenting the overview of the congestion control, the use of simple linear control functions in the end-to-end congestion control, and finally a basic analysis of the congestion control process within TCP was given. This analysis is essential for the implementation of Dynamic Bandwidth Partitioning in TCP/IP-based networks. This will be explained in more detail in the next Chapter, in which the detailed definition of the scheme will be given.

3.4. Bandwidth Allocation Concepts

3.4.1. Overview

The emerging multi-class Internet environment brings many new challenges to the research on bandwidth management in communication networks. As discussed is section 2.1, the major problem in the network design in the new environment is to guarantee appropriate levels of network performance to all traffic classes, while at the same time ensuring that the network is not under-utilised. This section analyses different approaches to bandwidth allocation in the new environment.

It is important to note that bandwidth allocation should be considered as a typical problem of resource allocation in the multi-class systems. The problem of sharing resources between competing demands has been much studied. The simplest policies, and in some sense extreme policies, are *complete sharing* and *complete partitioning*.

Complete sharing allows all services to share the resource indiscriminately. Complete sharing is the concept of the data-only computer network, in which the entire traffic is accepted in the network, and each transmission receives the equal part of the network's capacity. This concept is analysed further in section 3.4.2.

Complete partitioning divides the resource between the services, allowing each service exclusive use of its allocated capacity. This concept does not achieve maximal capacity utilisation. However, resource partitioning in the Internet proved to be useful for a number of applications, including the creation of virtual private subnetworks, in the mechanisms for advance reservation of real-time network services, and in the fast establishment of real-time connections [GUP95]. Section 3.4.3 analyses further the complete partitioning concept. Resource partitioning (in our case bandwidth partitioning) is the basic concept of the bandwidth allocation scheme presented in this Thesis. The objective of this Thesis is to show that a dynamic bandwidth partitioning scheme can improve the efficiency of the network, mainly in terms of satisfying the needs of end-users.

Section 3.4.4 presents possible concepts for the aggregation of the paradigms of the complete sharing and complete partitioning. Several interesting propositions are introduced, mainly concerning the use of the concept of Trunk Reservation. These ideas are relevant for the development and analysis of our scheme, since they present the different approaches researchers took to solve the same problem we are trying to solve.

3.4.2. Complete Sharing

The concept of complete sharing is the inherent concept of the data communication network. The network that implements a complete sharing scheme is said to provide best-effort treatment to its traffic. This paradigm was originally designed for data networks, in which the quality of service problem was not important. The most important application on such a network was simple file transfer.

In the current multi-class communication network, the quality of service has become very important. The principles on which the best-effort data network has been built prove to be insufficient for the QoS provision. Therefore, although complete sharing provides high bandwidth utilisation, a best-effort network requires improvements.

There are several propositions of how to improve the drawbacks of the best-effort scheme. Hurley and Le Boudec [HUR99] propose a simple scheme that does not introduce any additional complexity in the network, while at the same time providing excellent jitter results. Each best-effort packet is marked as either *green* or *blue*. Green packets receive more losses during bouts of congestion than blue ones. In return, they receive less delay jitter. No rate reservation is assumed. Traffic management and charging practices remain essentially the same as for the single class best-effort network.

Each packet is assigned a finishing service time deadline, a tag, and the packet currently having the lowest value is served first. Each green packet arriving is assigned a finishing service time deadline equal to arriving time t. A blue packet is assigned a time equal to the arrival time plus a constant D, namely $t+D$. The admission control comprises a modified version of Random Early Detection (RED) [FLO93] in which the dropping probability for blue packets is the usual RED dropping probability p while for green packets is αp. In addition, green packets must pass a second acceptance decision, to ensure that they are given a sufficiently small delay. It is shown in [HUR99] that an application that requires low delay jitter, and thus sends green packets experiences the low jitter, but at the expense of lower throughput.

On the other hand, it is widely accepted that all solutions for the bandwidth management in the future multiservice IP network will have to involve pricing mechanisms to prevent users from sending all their traffic in the highest priority class. The *Paris Metro Pricing* [ODL99] scheme relies on pricing alone to provide differentiated services. The PMP proposal is to partition the main network into several logically separate channels. In the basic design, each would have a fixed fraction of the capacity of the entire network. All channels would route packets using protocols similar to the current ones, with each packet treated equally. The only difference between channels would be that they would charge different prices. There would be no formal QoS guarantees, with packets handled on a "best-effort" basis. The expectation is that the channels with higher prices would be less congested than those with lower prices. There is an analysis of a simplified version of PMP by Gibbens, Mason and Steinberg [GIB] which shows that in their model PMP would be optimal for a monopoly service, but a carrier offering PMP would lose to a competitor offering an undifferentiated service. The author also underlines the importance of the ability to *assign varying capacities* to the separate channels, and also to vary prices for using those channels. That gives service providers substantially more flexibility than might appear at first.

The idea of partitioning the network and then dynamically changing the way the network is partitioned is very close to the concept of Dynamic Bandwidth Partitioning. However, in the PMP approach, the partitioning is introduced on the base of pricing for the services, while our approach is based solely on the level of network performance, without any explicit notion of the pricing mechanism.

In summary, the complete sharing paradigm provides room for high network utilisation, but does not guarantee the required performance levels. It is an excellent choice for the data-only networks and for the lightly-loaded parts of the global Internet.

3.4.3. Complete Partitioning

For real-time communication services to achieve widespread usage, it is important that network managers are allowed to control the services effectively. This means that it is very important to design the network in such a way that the network is able to guarantee the real-time applications will receive the performance level they require, in terms of the allocated bandwidth or the end-to-end delay and jitter. An important management capability concerns resource partitioning.

Resource partitioning is distributing the different resources available at any given network node or link among a number of partitions, where the admission control and establishment computations for a given connection need to consider only the connections in the same partition, and are completely independent of the connections accepted in other partitions.

Consider a network link with bandwidth B. The complete partitioning scheme decouples the link bandwidth into K independent 'sublinks'. The partitioning is defined with the set of partitioning parameters $\alpha_k, k = 1,...,K$. Each individual sublink k occupies the bandwidth $B_k = \alpha_k B$. If we know the utility functions for each of the traffic classes and the traffic intensity, it is possible, by using linear programming, to calculate an optimal set of partitioning parameters [ROS95], which maximise the overall utility on the link. Sublinks can be treated independently, therefore the performance measures of interest can be easily calculated.

The complete partitioning scheme is usually considered to be optimal for the network only when the traffic load is very heavy. This paradigm will never provide highest capacity utilisation but, observed in the multiservice environment of a network whose goal is to utilise its capacity in terms of maximising the revenue, quality of service and user's satisfaction, bandwidth partitioning can still be seriously considered as a resource allocation scheme. Furthermore, bandwidth partitioning is useful for the creation of *virtual private networks* (VPN). In the VPN environment the network capacity needs to be partitioned and leased to a number of users, who pay for the usage of a part of the network capacity. Bandwidth allocation in Virtual Private Networks is discussed in more detail in section 3.4.5.

A large amount of practical and theoretical work has been done in analysing the bandwidth partitioning concept. Very important work has been done by Floyd and Jacobson [FLO95] on the concept of *Link Sharing*. Link sharing allows multiple agencies, protocol families, or traffic types to share the bandwidth on a link in a controlled fashion. Because link sharing can be used to limit the bandwidth of traffic classes during times of congestion, link sharing isolates traffic classes from each other. This isolation can be an important mechanism for both accommodating emerging real-time application requirements, and for the protection of non-real-time traffic. The bandwidth on a link might be shared between multiple agencies, and each agency might want to share its allocated bandwidth between several traffic types. This leads to *Hierarchical Link Sharing*.

The link-sharing allocations can be either *static* (permanently assigned by the network administrator) or *dynamic* (varying in response to current conditions on the network, according to some predetermined algorithm).

The *static* resource management approach does not take into account several important issues, such as the dynamics of the communicating clients, the dynamics of the network state, and the trade-off between quality of service and network availability, thus affecting both the availability and the flexibility of the real-time network services. For example, in a lossless digital image browsing application, where degradation of image quality is not allowed, variation of the browsing speed corresponds to different requirements for network bandwidth. In addition, the static approach does not address the trade-off between QoS and network availability; higher QoS offered to a fraction of the clients may lower the availability of the network, and cause other communication requests to be rejected. Furthermore, once a link is partitioned, the capacity allocations remain fixed irrespective of the actual usage by the organisations. Thus, if one of the organisations is not fully using its capacity, the free link capacity cannot be used by other organisations to accept new real-time sessions.

Another interesting problem in resource partitioning is the routing aspect, i.e. the choice of the route for the traffic flows in the multi-class environment. Several authors have worked on that problem. Chen and Nahrstedt [NAH98] analysed the coexistence of the 'QoS' and 'best-effort' flows from the routing and scheduling point of view. They presented two sets of source routing algorithms: the bandwidth-constrained routing with imprecise state information for QoS flows, and the max-min fair routing for the best-effort flows.

Steenkiste and Ma [STE00] propose a routing algorithm that allows dynamic sharing of link resources among multiple traffic classes. In their multi-class routing algorithm, link resources are *dynamically* partitioned between QoS traffic and best-effort traffic. The idea behind their approach is to discourage QoS sessions from using links that already carry a heavy best-effort traffic load by increasing their link cost. This is achieved by using a link cost for QoS traffic that is based on the *virtual residual bandwidth* instead of the actual residual bandwidth. The virtual residual bandwidth captures the congestion conditions of best-effort, i.e. it is lower than the actual residual bandwidth on links that, relative to the rest of the network, carry a lot of best effort traffic and higher on links that have little best-effort traffic. The packet scheduler in a router or a switch ensures that the QoS traffic will at least get the amount of bandwidth that it reserved. Best-effort sessions use the bandwidth left unused by the QoS sessions. The algorithm uses the max-min fair share rate as a barometer of network congestion. This approach is interesting from the point of view of the Dynamic Bandwidth Partitioning scheme, since it shows that different kind of metrics can be used to inform the network control about the level of the performance. Where our scheme uses end-user utility, Steenkiste and Ma use virtual residual bandwidth.

3.4.4. Other Concepts, Virtual Partitioning and Trunk Reservation

In the previous two sections, two opposite concepts for bandwidth allocation were presented. This section analyses the major drawbacks of both concepts and analyses the direction for the further research in the area. As mentioned before, the complete sharing concept is perfect for data networks, which are unaware of the QoS issue. The complete sharing improves the network availability, fairness of allocation, and capacity utilisation, but does not provide any performance guarantees.

On the other hand, complete partitioning concept creates the environment in which traffic flows from each traffic aggregate/class are transported independently. This concept gives very high level of control to the network management, giving way to the design of powerful QoS-aware networks. However, the concept of complete partitioning suffers from low capacity utilisation and poor fairness, especially in the environment of very bursty heterogeneous traffic.

The question, then, is what do we need to do to design a network that contains the positive aspects of both of the presented concepts, while at the same time minimising the drawbacks of both of them. We broadly speak about the multi-class Internet environment, in which two major types of traffic, the real-time traffic and the elastic (data) traffic compete for the network capacity.

One possibility would be to consider the needs of the real-time traffic first, and to schedule the non-real-time traffic after the needs of the real-time traffic had been met, without having the scheduler to enforce bandwidth limitations on the real-time traffic. This scheme can be defined as extension for the complete sharing. After the needs of the real-time traffic had been met, link-sharing mechanisms would be used to share the remaining bandwidth among the non-real-time classes. However, Stoica et al [STO00] argue that this type of approach to link sharing is not sufficient, because it could lead to starvation of the non-real-time traffic over substantial periods of time. The second approach, presented by Altman et al [ALT97] consists of limiting the amount of bandwidth available for best-effort service categories, so that the best-effort traffic has some minimum pre-allocated bandwidth. This approach to the elastic traffic is similar to the one we are considering in our Dynamic Bandwidth Partitioning scheme.

The second possible concept would be the use of *trunk reservation*. Trunk reservation [KEY90] is based on limiting the capacity available for the low-priority traffic in the network in the case of congestion. An interesting idea based on the trunk reservation is *physical partitioning*, defined by Mitra and Ziedins in [MIT96]. In this scheme the capacity is subdivided between the competing services. Consider that we have K classes of traffic. Calls of class k are assigned capacity C_k. Each of these k trunk groups is assigned a *trunk reservation* parameter r_k, and calls of each class have priority at their allocated trunk group. A call arriving at its trunk group is accepted if there is spare capacity on it. If there is no spare capacity then the call chooses, in accordance with some discipline, another trunk group where the amount of spare capacity exceeds the trunk reservation parameter r_k. The simplest discipline picks the overflow trunk group at random. By allowing state-dependence the discipline for selecting the overflow trunk group may be generalised. However, this scheme shows some undesirable characteristics, since in some cases usage is even more unfairly distributed if some of the traffic classes arrive in bursts. Nevertheless, this concept of physical partitioning is very interesting, showing that we are able to create a partitioning scheme which is likely to improve the fairness in comparison with the complete partitioning scheme, mainly because it is able to adapt (up to the certain limit) to the dynamics of the traffic on the link/network.

In order to improve such a concept, Mitra and Ziedins continued their research to define a scheme called *Virtual Partitioning* [MIT96]. Virtual Partitioning is a bandwidth sharing scheme in which, instead of each class of traffic having a fixed priority, the priorities depend on the state of the system. In this scheme, the capacity is subdivided between the competing services. Let us assume that K traffic classes are using one communication link of capacity C. Traffic class k is assigned a nominal capacity C_k, where $\sum_{k=1}^{K} C_k \geq C$, and C_k is defined to be 'sufficiently high to ensure the desired grade of service'. Let n_k denote the number of calls of class k in progress. An arriving call of class k is always accepted if $n_k < C_k$ and $\sum_{j=1}^{K} n_j < C$. If $n_k \geq C_k$, the call is accepted if $\sum_{j=1}^{K} n_j < C - r_k$, where r_k is a trunk reservation parameter chosen to protect the remaining classes against overloads of class k. Thus, depending on whether $n_k < C_k$ or $n_k \geq C_k$, the priority status of class k is either high or low.

In [MIT97] the concept of Virtual Partitioning (VP) is extended *to Hierarchical Virtual Partitioning* (HVP). HVP is a concept for providing services to customers on a shared physical infrastructure. It is able to handle several such customers, where each customer sends and receives the traffic belonging to several traffic classes.

We will refer to customers as superclasses and the customers' services as subclasses. A single link with bandwidth or capacity C is shared between J superclasses, indexed by $j, 1 \leq j \leq J$, the j^{th} superclass having K_j subclasses. Let class (j,k) denote subclass k in superclass j, $1 \leq k \leq K_j$, $1 \leq j \leq J$. The j^{th} superclass is allocated a nominal capacity $C_j (C_j \geq 0)$ where $\sum_j C_j \geq C$. The $(j,k)^{th}$ class has nominal capacity $C_{jk} (C_{jk} \geq 0)$ allocated to it, with $\sum_{k=1}^{K_j} C_{jk} \geq C_j$.

Let n_{jk} be the capacity currently occupied by calls of subclass (j,k). Let $n_j = \sum_k n_{jk}$ be the capacity currently occupied by calls of superclass j. Finally, let $n = \sum_j n_j$ be the total capacity currently occupied. The HVP control policy is to admit a new call of class (j,k) if

$$n \le C - r_{jk} I\{n_{jk} > C_{jk} - d_{jk}\} - R_j I\{n_j > C_j - d_{jk}\} - d_{jk} \qquad (3.10)$$

where $I\{f\} = \begin{cases} 1, \text{if } f \text{ is true} \\ 0, \text{if } f \text{ is false} \end{cases}$, and d_{jk} is the capacity required by a call of subclass (j,k).

The expression in (3.10) means that the calls of overloaded classes are admitted only if there is unutilised reserved capacity, as in classical trunk reservation schemes. In contrast to the classical scheme, the trunk reservation here is dynamic and state-dependent. That is, the reservation level operating against a call of class (j,k) depends upon whether, if the call is admitted, only the subclass is overloaded or both subclass and superclass are overloaded. The corresponding reservation parameters are r_{jk}, R_j, and $r_{jk} + R_j$.

It is clear that all three schemes presented by Mitra and Ziedins bring the bandwidth allocation closer to the complete sharing concept. The link capacity in their schemes is fairly shared among active traffic flows, and only in the admission control strategy is there a difference between the high-priority and the low-priority traffic. While this makes the schemes very easy to implement in the real network, the question is how would these schemes perform in the real-network environment, which needs to be modelled with a more accurate model then the one used in [MIT96] and [MIT97], which was briefly explained here. This mainly refers to the fact that in the above model, all calls have equal capacity requirement.

The concepts that bring the bandwidth allocation closer to the complete partitioning concept are somehow different. It is obvious that the static nature of the complete partitioning scheme is the most important drawback of that idea. In the static link sharing concept presented in [FLO95], the capacity of a link is statically divided between QoS and best-effort sessions. The problem there is how to determine what fraction of the link capacity should be used for each traffic class because the ratio of QoS sessions over best-effort sessions changes over time. A *semi-dynamic* link sharing policy may be employed to overcome the problems of static link sharing. Under this sharing policy, the measured *link utilisation* for different traffic classes is used to periodically update what fraction of the link capacity is assigned to each traffic class.

While this provides some degree of adaptation, this strategy may not work well if there are sudden changes in the utilisation. Often these changes are difficult or impossible to predict. Furthermore, the link utilisation is not always a good indicator of how much bandwidth is available for reservation. Instead of the link utilisation, other performance measurements can be used. These issues will be further analysed in Chapter 4, where our Dynamic Bandwidth Partitioning scheme will be analysed.

3.4.5. Bandwidth Allocation in Virtual Private Networks

A very important issue in the analysis of the bandwidth allocation in multi-class Internet is the analysis of Virtual Private Networks (VPN).

It is well known that many businesses cannot survive in the modern world without reliable telecommunication networks, which should simultaneously provide high levels of availability and security. Business users are likely to require a large set of applications from the Internet, ranging from plain telephone connections and open access to the World Wide Web, to sophisticated video conferences and highly reliable connections for data transfer. Network providers need to design highly reliable and robust network solutions which can satisfy these sophisticated requirements. Furthermore, business people travelling in remote parts of the world expect the same access to the corporate LAN as their colleagues sitting in the office. Remote office sites expect a secure inter-connection which will speed up the company's business. This increased need for quality of service, new sophisticated applications and communication privacy drive the recently very popular concept of Virtual Private Networks (VPN).

A VPN can be defined as a private network constructed within a public network infrastructure, such as the global Internet. VPNs use dedicated secure paths, or tunnels, for transporting data between remote users and the corporate LAN. The goal of virtual private networks is to provide customers with the quality of service that approximates as closely as desired that experienced on dedicated, private networks, while, in fact, the aggregate traffic is multiplexed on a shared network.

The object of the resource management schemes in the VPN environment should be to realise multiplexing gains and to maximise the utilisation of available network resources. A key feature of any VPN from the point of a resource manager is the difference in the mode of resource sharing that exists at the two levels, one at the level of customers and the other at the level of the various services of each customer.

The majority of the theoretical work done recently on the performance enhancement of VPNs concerns security. Much less attention has been paid to the issue of resource management, both in the core Internet that serves multiple VPNs, and in the individual VPNs.

This Thesis is interested in bandwidth allocation issues in VPNs. This section analyses the optimal bandwidth allocation which can achieve maximum performance from the concept of virtual private networks – both in terms of the user satisfaction and in terms of the network utilisation.

In general, the implementation of VPNs from the point of the bandwdith allocation is the natural extension of link sharing to VPNs. The service provider leases bandwidth on a path to an organisation. The organisation tunnels its VPN traffic through this path as if it were a single virtual link. Such Virtual Leased Links (VLLs) may be realised by tunneling, Label Switched Paths in an MPLS network [RFC3031], or Virtual Paths in an ATM network [ONV94]. We are interested in designing the way to allocate bandwidth to active VPNs, so that the bandwidth allocation generates maximal users' utility (or maximal level of the network performance).

The most obvious solution to the problem would be static link sharing (complete partitioning) of the available bandwidth. This solution suffers from the same shortcomings that were analysed in the previous sections. The traffic cannot be sent on a virtual link at a rate higher than its allocated capacity, even if the underlying provider network has large spare capacity. Thus the resource sharing in the provider's network cannot be achieved without dynamically changing the capacity of virtual links. There are several works that analyse this problem further and present a more dynamic solution.

Goyal et al presented in [GOY99] a comprehensive study of an IP telephony network, and present a *Distributed Open Signalling Architecture* (DOSA) as the solution. For us, the resource management part of this architecture is particularly relevant. In order to support large-scale services, the DOSA architecture assumes that admission control and scheduling must be used in the backbone network on an *aggregate basis*. In other words, phone calls with the same source and destination IP address should be aggregated and a certain level of bandwidth should be allocated to these aggregates. The level of allocated bandwidth is periodically *resized*. This resizing is driven by a prediction of the number of voice flows expected to be carried on the aggregate in the future. The simulation results show that resizing bandwidth levels on a large-scale IP telephony network once every 5 minutes saves a factor of about 2 in capacity on all links. This architecture, although designed for transporting voice traffic only, shows the benefit of the dynamic resizing of bandwidth partitions.

Duffield et. al. [DUF99] analyse heterogeneous traffic, consisting of voice and data calls, which is more similar to the traffic served in the Virtual private networks. They point out the need for aggregating the traffic that belongs to a specific VPN and travels to a specific destination in the network. They call such traffic aggregates *hoses.* The capacity in the core network is reserved for the *hoses*, where that capacity needs to be dynamically allocated, because of the uncertain profile and volume of the traffic within each VPN. To manage resources so as to deal with this increased uncertainty, they consider two basic mechanisms: *Statistical Multiplexing* and *Resizing.*

As a single QoS assurance applies to a *hose*, the provider can consider multiplexing all the traffic on a given hose together. Similarly, the set of hoses making up the VPN have a common QoS assurance, and the provider can consider multiplexing all the traffic of a given VPN together. These techniques can be applied on both access links and network internal links. Providers can take the approach of allocating the capacity for the traffic on a VPN statically, taking into account worst case demands. Alternatively, a provider can make an initial allocation, and then resize that allocation based on online measurements. Again, such techniques can be applied on both access and network internal links. Simulation results presented in [DUF99] show that the capacity requirement gain is around 2 for statically provisioned capacity over the basic, customer-pipe approach, where each flow reserves the required capacity. Dynamic resizing of the allocated capacities for traffic aggregates (hoses) introduces additional gain factor of around 2.

Similarities and differences of these schemes to our Dynamic Bandwidth Partitioning scheme will be explained in more detail in the next Chapter. For now, it is important to underline that the dynamic bandwidth resizing proposed in [GOY99] and [DUF99] is done periodically, based on the long-term traffic predictions. Our scheme assumes non-periodic bandwidth resizing, where the resizing happens as a consequence of feedback information about the network state.

Another important issue is the *fairness* of the capacity allocation between a number of active VPNs in the network. This problem was analysed by Garg and Saran [GAR00]. They present the *Stochastic Fair Sharing* scheme in which the priority of a VPN depends on the normalised usage of a fraction of a link allocated to the VPN. They claim that a good alternative to the static link sharing concept is to fairly redistribute the free capacity by resizing depending upon the current usage of different organisations sharing the link. Without the fairness guarantees, a few virtual links may take most of the bandwidth of the network which can result into starvation of other virtual links. They propose a scheme *called Stochastic Fair Sharing* (SFS) to implement fair sharing among virtual links by dynamically modifying their capacities. Virtual link capacities are increased upon session arrivals and are decreased as sessions complete. As a result, the redistributed capacities, which are available for session lifetimes, can be used by the real-time traffic as well as the best-effort traffic with minimum bandwidth requirement.

To ensure fairness and protection, the SFS admission control algorithm decides whether increasing the capacity of a logical link is acceptable. Once a new session (flow) is accepted, the corresponding weights in the underlying packet scheduler are adjusted to reflect the change in the bandwidth allocation.

The SFS scheme prioritises the VPNs on the basis of their *normalised usage*. Let r_i be the amount of the capacity already reserved by logical link i (or VPN i), and let c_i be the capacity allocated to the logical link i. The normalised usage of logical link i is given by $n_i = r_i / c_i$. The logical links are labelled in the increasing order of their normalised usage, so that link 1 has the lowest normalised usage, and link N has the highest. Therefore, the link with the lowest normalised usage is given the highest priority. The consequence of this is that this scheme gives higher priority to sessions of logical links having low normalised usage and thus attempts to equalise the normalise usage of different logical links. In general, a session of logical link i is accepted if the free capacity after accepting the session is at least equal to the sum of the trunk reservations of the logical links with lower normalised usage. Formally, a new session of logical link i, with bandwidth request of r is accepted if and only if

$$\sum_{j=1}^{N} r_j + r + \sum_{j<i} t_j \leq C \qquad (3.11)$$

Where r is the reservation requested by new session, and t_i are the trunk reservation parameters. Furthermore, if the normalised usage of a logical link is close to its fair share, then it is not necessary to have a large value of trunk reservation for the logical link. In this case, we reduce the trunk reservation of the logical link. Logical link i has a static trunk reservation parameter $\overline{t_i}$ which is the maximum value of trunk reservation for the link. The fair share f_i for each logical link is dynamically computed. If the difference between the fair share and the current reservation of a logical link is less than its static trunk reservation, then the trunk reservation for the link is set to the difference. Otherwise, the trunk reservation is set to the static value. Formally, $t_i = \min\left[\overline{t_i}, f_i - r_i\right]$. The fair share of the logical link is computed by redistributing the free capacity of logical links with lower normalised usage as follows:

$$f_i = \frac{C_i}{\sum_{j=1}^{N} C_j}\left(C - \sum_{j=0}^{i-1}\left(r_j + t_j\right)\right) \qquad (3.12)$$

If the usage of logical link is less than its fair share, then the free capacity is fairly redistributed among heavily loaded links after keeping aside a small trunk reservation.

The Stochastic Fair Sharing scheme provides high throughput and low blocking probability because of better statistical multiplexing and sharing. It is very similar to our Dynamic Bandwidth Partitioning scheme, as will be explained in the next Chapter.

3.5. Summary

This Chapter studied bandwidth allocation in the multi-class Internet. Majority of the work done previously in the bandwidth allocation for IP networks was done for a network which serves a single type of traffic. The objective was usually to share the bandwidth to maximise the link utilisation, while achieving certain level of fairness. This Chapter analysed different forms of fairness, and explained the new multi-class Internet environment.

The main feature of the multi-class network is the heterogeneity of the traffic. In the new environment, a new approach to fairness is needed. End-users of real-time applications require *performance guarantees*; it is necessary that those guarantees are included in the way we observe and evaluate fairness.

The second important issue discussed in this Chapter is congestion control. Congestion control is a powerful network mechanism that enables the network to control the amount of incoming traffic, thus controlling the network performance. End-to-end congestion control using linear control algorithms is especially important for our research since our Dynamic Bandwidth Partitioning scheme, which will be presented later in the Thesis, is based on a linear control mechanism. Congestion control in TCP is implemented using additive increase, multiplicative decrease linear control algorithm which enables the network to inform the end-users to adapt their sending rate according to the congestion information. The TCP congestion control mechanism has enabled TCP/IP-based Internet to overcome the congestion collapse. However, congestion control alone is not sufficient to provide a full end-to-end quality of service network. Additional mechanisms and controls are needed.

Section 3.4 of this Chapter analysed some of these mechanisms, namely the ways that the network capacity (bandwidth) can be allocated and shared among numerous traffic connections. Two basic bandwidth allocation concepts were discussed: complete sharing and complete partitioning of bandwidth. Each of these two extreme concepts is designed to favour certain applications. Complete bandwidth sharing is the optimal concept for file-transfer applications, where high utilisation is more important then performance guarantees. On the other hand, complete bandwidth partitioning strictly separates the traffic belonging to different traffic classes. This concept paves the way for the introduction of the performance guarantees, but is inelastic and therefore efficient only for the high traffic loads.

The optimal solution for the multi-class network lies somewhere between these two extreme concepts. There is a large amount of theoretical work on the optimisation of the resource allocation in a multi-class environment similar to the present Internet. Some of the concepts were presented in section 3.4. The analysis given there is the introduction to the next Chapter, which presents the main contribution of our research. Chapter 4 defines in detail a new bandwidth allocation scheme, which uses the concept of complete bandwidth partitioning, but introduces dynamics to this rigid concept and thus proves to be efficient for a large spectrum of traffic load scenarios.

CHAPTER 4

DYNAMIC BANDWIDTH PARTITIONING

4.1. Introduction

This Chapter presents a new bandwidth allocation scheme for the multi-class Internet. The scheme is called Dynamic Bandwidth Partitioning (DBP). The key feature of this scheme is that the available bandwidth is *partitioned* between active traffic classes, so that traffic forwarding of each of the traffic classes is independent. Furthermore, the partitioning is *dynamic*, adapting to the state of the links in the network in order to generate maximal end-users' utility.

This Chapter analyses the Dynamic Bandwidth Partitioning scheme in detail. Firstly, section 4.2 presents the utility-based approach to resource optimisation, and the overview of the previous research done on optimising resource allocation to maximise the generated utility. Since this is a very large theoretical problem, we focus on solving the problem of capacity (bandwidth) allocation in IP-based multi-class communication networks.

Section 4.3 presents the Dynamic Bandwidth Partitioning scheme in detail. We start with the scheme overview, where the main issues are introduced. Each of these issues is later addressed individually: admission control, bandwidth allocation, and the partitioning algorithm.

The important part of our research is concerned with the implementation of the DBP scheme. Section 4.4 discusses the implementation issues, including the extensions to the partitioning algorithm, and the guidelines for the implementation of Dynamic Bandwidth Partitioning in MPLS networks and in the VPN network environment. Section 4.4.4 analyses further the features of the MPLS network and presents the ongoing work of dynamic renegotiations of Label Switched Paths in MPLS networks. There are obvious analogies between dynamic negotiation of the LSPs in the MPLS network and the Dynamic Bandwidth Partitioning scheme. These analogies are studied in more detail. Section 4.4.5 presents a Hierarchical DBP scheme, which can be used for the efficient resource allocation in the VPN environment.

4.2. Utility-based Resource Allocation

4.2.1. The Problem Formulation

The notion of utility has already been discussed in sections 2.2.3 and 3.2.2. Here we define the utility-based resource allocation problem. We start from the fact that each end-user of an Internet application derives certain utility from the application he is using. We can treat utility as a measurement of the level of end-users' satisfaction; it is a subjective measure of the application performance. Naturally, utility depends on the quality of the service given by the network. Traditionally the quality of the network performance has been measured in a number of ways, using end-to-end delay, delay jitter, throughput, etc. By using utility, we are trying to find a universal metric for the end-user satisfaction.

If we want to design the optimal utility-based resource allocation, we should design the resource allocation that maximizes the mean utility end-users derive from the network. This is exactly what Kelly argues in [KEL97], that bandwidth should not be shared so as to maximise the link utilisation, but rather to maximise an objective function representing the overall utility of the flows in progress.

It is important to understand that throughout this Thesis we assume that the resource we are allocating/partitioning/sharing is the link bandwidth, and that the derived utility for the end-users mainly depends on the amount of bandwidth being allocated to the application that end-user is using. Naturally, this assumption is not completely correct, since in many applications other resources or performance parameters are more important for the end-users' satisfaction, such as CPU time or buffer space.

The measurement of the utility is a problem of its own. In theory, each individual user derives its own utility from the network performance. A very good analysis of the variability in end-users perceptions of the network performance is given in [BOU00]. Utility levels are calculated using *utility functions*, which have already been mentioned a number of times in this Thesis. Utility functions map network parameters (delay, throughput, packet drops, etc) into the performance of an application; they reflect how the performance of an application depends on the network parameters. The current traffic heterogeneity in the Internet brings numerous different applications with hundreds of different utility functions. It is possible that every single user of an Internet application has a different utility function.

Therefore, we need a scheme which allocates the available bandwidth in a way that maximises the utility experienced by end-users. More formally, let us introduce the analytical model for the utility-based resource allocation. Let us consider a network link of capacity C, which serves heterogeneous Internet traffic. Each traffic flow j is allocated amount of bandwidth x_j, and defines its utility function $U_j(x_j)$. The goal of the utility-based resource allocation is to find the optimal bandwidth allocation x_j, so that the overall utility $U = \sum_j U_j(x_j)$ is maximised.

This is a multi-constrained optimisation problem. It is obvious that the number of different utility functions is the same as the number of different traffic flows, which makes the problem very complicated. Designing a bandwidth allocation scheme which takes into consideration all these individual utility functions would be complicated, if not impossible to do. The scalable solution would be to differentiate the traffic into I traffic classes, and to assume that each traffic class has only one utility function defined, $U_i(x_i)$. Now the optimisation problem can be defined as finding the optimal bandwidth allocation x_{ij} for each traffic flow j belonging to traffic class i, so that the overall utility $U = \sum_i \sum_j U_i(x_{ij})$ is maximised.

There is a massive amount of theoretical work in operations research, economics and resource allocation behind the solution for such a problem. In this Chapter, we will focus on solving the practical problem of bandwidth allocation in the multi-class TCP/IP network. We will study the theoretical optimal solution to the utility-based resource allocation, and present a simple scheme that provides an increase in the overall utility level, compared with allocation schemes with no control.

The next section brings the survey of the most important research done on the use of utility in the resource allocation for packet networks. It is followed by the detailed explanation of the approach we are taking.

4.2.2. Overview of the Related Work

The problem of allocating available resources to satisfy multiple requirements is very complicated and a significant amount of work has been done on solving the general resource allocation problem. The domain of operations research has particularly paid attention to solving such a problem. In this Thesis we are interested mostly in one particular case of the resource allocation problem, which is allocating a single resource (bandwidth) among a number of users that are differentiated into classes of users, each class responding differently to the way bandwidth is being allocated. In this section, we choose from the massive amount of work that has been done the fraction that considers our particular problem. All of the approaches analysed here have a common objective of maximizing the network performance in terms of the users' utility.

The starting point is the argument of Kelly et al [KEL97] that bandwidth should not be shared so as to maximise the link utilisation, but rather to maximise an objective function representing the overall utility of the flows in progress. This argument is followed by Key and McAuley, who in [KEY99] present a simple network model which is very relevant to our problem. Suppose each user r has a utility function $U_r(x_r)$ relating bandwidth to 'worth', where x_r is the traffic rate for user r. The utility function is assumed to be *concave.* Key and McAuley analyse the equilibrium point where 'social optimum' and 'user optimum' coincide. Social optimum is defined as the overall utility, $\sum_r U_r(x_r)$, decreased by some cost, C. The user optimum is the experienced utility, $U_r(x_r)$, decreased by the price the users pay for using the network, $\alpha_r x_r$. Kelly et al proved in [KEL98] that, if the utility functions are $U_r(x_r) = \omega_r \log x_r$ (meaning they are concave and differentiable), the allocation that provides the social optimum generates *proportional fairness*. His general conclusion in [KEL97] was that if each user is able to choose a charge per unit time that it is prepared to pay, and if the network determines allocated rates so that the rates per unit charge are proportionally fair, then a system optimum is achieved when users' choices of charges and the network choice of allocated rates and prices per unit share are in equilibrium.

Although Kelly, and the other authors mentioned, analyse the optimal congestion pricing in the single-service system, and they use only concave utility functions, their work is still very relevant to our research. The analysis and optimisation they provide are general enough to be implemented in the case of the multi-class networks which we analyse.

Cao and Zegura [CAO99] introduced *utility max-min* fairness. They start from the fact that a pure best-effort network service such as in IP networks provides no feedback to adaptive applications. In this environment, the traditional max-min bandwidth allocation will often result in significant disparity in the application-layer performance of applications, despite a "fair" allocation of bandwidth. The bandwidth max-min flow control mechanism attempts to maximise the bandwidth allocated to each application, with no regard to differences in applications. Cao and Zegura underline the need for the application-oriented approach to fairness, and define the utility max-min fairness as:

A feasible bandwidth allocation vector R is utility max-min fair if for each connection i, its utility $f_i(r_i)$ cannot be increased while maintaining feasibility, without decreasing the utility $f_j(r_j)$ for some session j which satisfies $f_j(r_j) \leq f_i(r_i)$.

This definition of fairness is very similar to the definition of max-min fairness (see Chapter 3), with the only difference that in this case the utilities on the bottleneck are equal, not the allocated capacity. The natural consequence is that the utility max-min allocation is equivalent to the bandwidth max-min allocation when the utility functions of all applications is equal. This is the case for the single-class Internet only.

Similar to the bandwidth max-min allocation, there are generally two ways to implement a utility max-min fairness policy in the network. One option is that all network switches use a utility-aware Fair Queuing scheduler. The other option is to have routers explicitly compute the max-min fair rate for each flow and periodically inform each source of its rate. In [CAO99], a simple algorithm of 'progressive filling' is defined, similar to the one for the bandwidth max-min allocation given in [NAH98] and [LEB].

The notion of utility max-min fairness is very important for our work, since our Dynamic Bandwidth Partitioning provides application-oriented resource allocation, keeping in mind the application requirements expressed through the utility functions. Naturally, in deploying bandwidth allocation schemes that generate utility max-min fairness, the correct choice of utility functions is essential. Cao and Zegura underline one more important issue in [CAO99], when they state that to design a feasible utility-based bandwidth allocation scheme, we have to trade-off the generality and simplicity in the specification of utility functions. The problem lies in the subjective nature of the utility itself. Each user could define its own utility function for some application. Optimising the network control to maximise the utility for the near infinite number of utility functions thus generated, could prove to be impossible. Therefore, some approximations are necessary. The approximation we are making here is that all users of the traffic belonging to a particular traffic class have equal utility functions. This will be explained in more detail later in this Chapter.

Very important for the development of our work is the work of Shenker and Breslau [SHE98]. They used a simple fixed load model to analyse and compare the reservation-capable and best-effort network architecture on the basis of the utility these schemes generate. Two major types of applications were analysed in their work – rigid and rate-adaptive applications. The mathematical expressions of utility functions for these application types are very relevant for us, especially for the case of the rate-adaptive applications.

Shenker and Breslau use their model to evaluate the increase in utility after network admission control is introduced. If that increase proves to be small, then there is no reason to add complexity to the network when the outcome is just a slightly more efficient network. Therefore, any control that is introduced to the network needs to be fairly simple. One of the ways they assessed the significance of the network control was to determine how much additional bandwidth was needed to make a best-effort only network have the same performance as the reservation-capable one. Their model showed significant performance and bandwidth gaps between best-effort and reservation-capable networks with rigid applications. Considering adaptive applications changed the picture dramatically. Their results show that the answer to the original question whether to deploy admission control and bandwidth reservation or not, depends in large part on the load patterns in the future Internet.

An interesting definition of utility functions for video traffic was presented in the work of Sarkar and Tassiulas [SAR99]. They presented an algorithm for computation of maximally fair utility allocation, and used a stepwise utility function for video applications. Each added layer of bandwidth to the video traffic flow increased the utility for a certain fixed value. It is interesting to note that such utility function is not strictly increasing. They studied fair allocation of utilities, where utility of a bandwidth could be the number of layers of the bandwidth itself, or any other function of the bandwidth depending on system requirements.

Very interesting work was done by Lee, Rajkumar et al [LEE99][RAJ98][RAJ97]. They presented a QoS management framework that enabled quantitative measurement of quality of service, analytical planning and allocation of resources. End users' quality preferences were considered when system resources were distributed across multiple applications such that the total utility that accrues to the end-users is maximised.

In [RAJ97] the authors presented an analytical model for QoS management in systems which must satisfy application needs along multiple dimensions such as timeliness, reliable delivery schemes, cryptographic security and data quality. They referred to this model as *Q-RAM* (QoS-based Resource Allocation Model). The goal of the model is to be able to allocate resources to the various applications such that the overall system utility is maximised under the constraint that each application can meet its minimum needs. Their model is very relevant for us, because it is based on the notion of *application utility*, which is defined to be the value that is accrued by the system when an application is allocated a certain amount of resources.

The model is rather general, assuming both multiple resources, and multiple QoS dimensions. The model assumes m resources $\{R_1, R_2, ..., R_m\}$. Each application needs to satisfy requirements along the multiple QoS dimensions. For each application i the application utility $U_i = U_i\left(R^i\right)$ is defined, where $R^i = \left(R_{i,1}, R_{i,2}, ...R_{i,m}\right)$ is the vector of allocated resources. Further to that, the model defines the *dimensional resource utility* $U_{i,k}$ of the application i, which defines the utility accrued concerning only one QoS dimension k. The QoS dimensions involved may be timeliness, data quality, reliable packet delivery, security achieved through cryptography, etc.

Very important from their work is the way the application utility function was derived. The application utility U_i is not necessarily equal to the sum of $U_{i,k}$. In [RAJ97] the authors gave an interesting analysis of the ways the application utility is composed from multiple dimensional utility functions. This is very interesting for our work, because it shows that there are ways to compose a single utility function even though there are multiple QoS requirements.

Another interesting part of the work in [RAJ97] is the resource allocation algorithm that is presented for the case of a single resource. The algorithm determines the optimal resource allocation for obtaining maximal utility for all active applications. Assumptions made for this algorithm are as follows. Firstly, there is only a single resource and a single QoS dimension. Secondly, the application utility functions U_i are assumed to be twice continuously differentiable and concave. They defined a theorem that provides a necessary condition for the resource allocation to be optimal. We give the theorem here in full.

Theorem. A necessary condition for a resource allocation to be optimal is $\forall i$, $1 \le i \le n$, $R_i = 0$, *or for any* $\{i, j\}$ *with* $R_i > 0$ *and* $R_j > 0$, $U_i^{'}(R_i) = U_j^{'}(R_j)$.

The proof of the theorem is given in [RAJ97]. Based on this theorem, the authors presented a simple algorithm of “progressive filling” which allocates available resource to maximise the overall utility. The algorithm is similar to the algorithm given by [NAH98] and [LEB] for achieving max-min fairness, only in this algorithm the utility, not the amount of allocated resource is maximised, in a similar way to [CAO99]. Although this is an *explicit rate* algorithm, which assumes a centralised intelligence which periodically calculates the optimal rates for the sources, it is still interesting for our work, because it presents the resource optimisation based on the *utility*.

In [RAJ98] and [LEE99] Rajkumar et al extend their work on the Q-RAM model. In [RAJ98], they show that the Q-RAM problem of finding the optimal resource allocation to satisfy multiple QoS dimensions which are not independent is NP-hard [GAR79]. Furthermore, they study the problem of allocating multiple resources to satisfy a single QoS dimension. Finally, in [LEE99], they analyse discrete QoS dimensions and relax the assumption about the concave utility functions. Since the QoS management optimisation problems are NP-hard, there are no optimal solution techniques other than a (possibly complete) enumeration of the solution space. On the other hand, QoS management calls for on-line solutions as the optimisation module will ideally be at the heart of an admission control and adaptive QoS management system. Therefore the goal is to strike the right balance between solution quality and computational complexity.

In summary, the work done by Rajkumar et al is very important for us. The most relevant parts of their work include the mechanism for deriving the single utility function from multiple

QoS dimensions, the stated theorem and the progressive filling algorithm for optimal allocation of a single resource to satisfy single QoS dimension. Finally, their work on proving that allocating multiple resources to satisfy multiple QoS dimensions is NP-hard paved the way for us to try to find near-optimal solutions which would be easy to implement, instead of analysing the possible optimal solution without taking the complexity under consideration.

Another interesting analysis is given by Banchs and Denda in [BAN00]. They presented a new architecture for relative service differentiation, which they call *Scalable Share Differentiation (SSD)*. This scheme allocates network resources on a user basis and properly provides isolation for elastic and real-time traffic, based on the contracts users made with their ISPs. Isolation is provided by the fact that resources are allocated on a user basis through *user shares*. Each user i contracts a share S_i with the network operator. This share is used to determine the treatment of the user's packets in bottleneck nodes, both for the elastic and real-time traffic types. Whenever there is not enough bandwidth for that traffic type to serve all incoming packets, the packets with a lower effective share should be dropped with higher probability.

Especially interesting for us is the fact that the SSD architecture *decouples* the two traffic types by using two queues for each outgoing link, one low priority queue for elastic traffic and one high priority for real-time traffic. With the queuing mechanism presented, elastic and real-time traffic are *separated* in such a way that the capacity of the link has to be divided among them. A key problem is how to *choose* the right proportion of the available capacity to be assigned to each traffic type. This problem is very important for the Dynamic Bandwidth Partitioning scheme.

The idea of user contracts and QoS negotiation is further developed in Abdelzaher et al [ABD97]. They proposed a model for QoS negotiation in building real-time services to meet both predictability and graceful degradation requirements. QoS negotiation is shown to (i) outperform conventional "binary" admission control schemes (either guaranteeing the required QoS or rejecting the service request), and (ii) achieve higher application-perceived system utility. The mechanism permits clients to express in their service requests a spectrum of QoS levels they can accept from the provider, and the perceived utility of receiving service at each of these levels. This basically means that the clients are free to define their utility functions. In this case, the negotiation model is centred around three simple abstractions: *QoS levels*, *rewards*, and *rejection penalty*.

A client requesting service specifies in its request a set of negotiation options and the penalty of rejecting the request derived from the expected utility of the requested service. Each negotiation option consists of an acceptable QoS level for the client to receive from the provider and a reward value commensurate with this QoS level. The service provider is free to switch the QoS level to another level in the client's negotiation options, if that increases the perceived utility.

By offering QoS degradation as an alternative to rejection, and by using admission control rules, the authors show that the reward sum (or perceived utility) achieved with their scheme is lower bounded by that achieved using conventional admission control schemes given the same schedulability analysis and load sharing algorithms. Their analysis is very interesting because of the notion of user-controlled QoS negotiation. However, it does leave questions on the implementation issues. Leaving each individual user to decide and negotiate its own utility function and QoS levels raises a problem of scalability.

Utility-based resource allocation has not been analysed only for IP-based telecommunication networks. For example, Jones et al [JON95] investigate algorithms for modular distributed real-time resource management in computer operating systems. The goals of their research include developing appropriate real-time programming abstractions to allow multiple independent real-time programs to dynamically coexist and share resources on the same hardware platforms. The design carefully separates mechanisms and policies, allowing varied or dynamic resource management policies to be used without modifying applications. They intended to use this flexibility to implement user-centric, rather than application-centric, resource management policies. It is clear that the general problem they try to solve is the same as for other authors mentioned in this section.

In summary, this section has reviewed briefly a number of works that focus on optimising resource allocation to generate maximal end-user utility. This choice of authors is presented here because each of the works mentioned has contributed to the development of our Dynamic Bandwidth Partitioning scheme. There is a common thread in all the ideas presented here – how to design an optimal bandwidth allocation scheme which can provide us with performance guarantees for the applications that require them, and with fairness for the data applications. Particular note is taken of strategies which involve heterogeneous traffic classes.

4.3. Dynamic Bandwidth Partitioning

4.3.1. Overview

Dynamic Bandwidth Partitioning scheme partitions the available bandwidth on the link between pre-defined traffic classes, so that the forwarding of the traffic from each class is independent. Furthermore, the scheme provides a simple linear control algorithm that dynamically modifies the bandwidth partitioning to achieve maximal end-users' utility.

The main features of the scheme have already been analysed in the Thesis – bandwidth partitioning, linear control, utility and utility functions, optimisation of resources.

Bandwidth partitioning in general makes traffic from different traffic classes independent, and therefore protects high-priority traffic from the effect of sudden burstiness of the low-priority traffic. Previous analyses of the bandwidth partitioning concept had a common conclusion that it is ineffective, due to low network utilisation. It was considered to perform better than the best-effort scheme only in the environment of very high traffic loads. A novel approach to bandwidth partitioning is presented here, showing that an adaptive partitioning scheme can perform better then the best-effort scheme even for moderate traffic loads. Our scheme is user-oriented, designed to generate maximal user-utility on the network.

The most important features of the scheme will be analysed in more detail in the rest of this section. These features are:

- Admission Control
- Bandwidth Allocation
- Partitioning Algorithm
- Implementation

4.3.2. Admission Control

Admission control consists in refusing an incoming flow if the acceptance of that flow will *unacceptably* affect the level of the network performance of other traffic flows in the network. Therefore, the most important question for the admission control procedure is whether the *overall* network performance is increased or decreased after the acceptance of the new flow. It is very difficult, however, to measure the level of the network performance.

The question whether to deploy an admission control mechanism in IP networks is one of the most important issues that has been analysed in a number of works [SHE98][ROB99][BRE00][VDB00]. Opponents of the admission control and reservation of bandwidth contend that a reservation-capable network will not deliver satisfactory service unless its blocking rate is low, and at such provisioning levels best-effort networks will provide a completely adequate service – service that is nearly as good as that of the reservation-capable network. On the other hand, there are arguments in favour of admission control. Sophisticated real-time applications require performance guarantees from the network, and the network without admission control at its boundaries in not likely to be able to provide these guarantees. Also, in the absence of admission control the ineffective traffic due to the retransmission of lost packets constitutes a significant overhead, since in TCP the packet loss rate increases as the capacity taken by the flow (bandwidth share) decreases.

When it comes to standardised QoS-aware Internet architectures, the Integrated Services architecture [RFC1633] introduces per-flow admission control. This architecture uses a signalling protocol to establish reservations at all routers along the path. While providing excellent QoS, this approach has a serious scalability problem, because it requires routers to keep per-flow state information and to process per-flow reservation messages. The Differentiated Services architecture [RFC2475] requires no per-flow admission control or signalling, and is therefore considered to be the solution for the scalability problems of the Integrated Services architecture. However, what Differentiated Services does not provide is strict performance guarantees. It can only provide qualitative QoS guarantees.

There are three basic mechanisms for admission control:

- Static Admission Control (SAC)
- Measurement-base Admission Control (MBAC)
- End-point Admission Control (EAC)

Static Admission Control admits new flows on the basis of an *a priori* description of the traffic characteristics, where the traffic is usually described by parameters such as peak traffic rate or maximum burst size. On the other hand, Measurement-based Admission Control bases its decisions on the measured state of the network. Finally, in the Endpoint Admission Control, the end host probes the network by sending probe packets at the data rate it would like to reserve and records the resulting level of packet losses. The host then admits the flow only if the loss percentage is below some threshold value. Endpoint admission control uses the regular best-effort infrastructure and, by adding control algorithms at the endpoints, provides performance guarantees to traffic connections.

In the Dynamic Bandwidth Partitioning scheme, admission control is introduced for all traffic classes, based on the minimal bandwidth requirement that the traffic flows have. We assume that all traffic classes, either elastic or real-time, put some minimal requirement on the network. Some of the traffic classes show adaptive features, and are able to adapt to any allocated bandwidth *above* the minimal required, and some are *brittle*, non-adaptive. The non-adaptive traffic classes usually put larger bandwidth requirement on the network. The minimal bandwidth level for the elastic (data) traffic is usually very small.

In general, if we consider a communication link of capacity C, and the incoming flow belongs to the traffic class i, requiring minimal bandwidth $b_i^{\min}$, the admission control request is as follows:

$$b_i^{\min} \geq C - \sum_j n_j b_j^{\min} \qquad (4.1)$$

where n_j is the number of active traffic flows of class j, and $b_j^{\min}$ is the minimal bandwidth required for the flows belonging to class j.

4.3.3. Bandwidth Allocation and Partitioning Algorithm

We need to find a way to allocate the bandwidth on the network link in a way which maximises the generated utility on that link. Furthermore, the bandwidth allocation needs to be simple, dynamic and compatible with the existing Internet protocols and the congestion control mechanism.

Consider a network with K links, where each link $k = 1,...,K$ has bandwidth B_k. Each traffic flow j occupies amount of bandwidth b_{jk} on the link k. A feasible bandwidth allocation $b_{jk} \geq 0$ is defined by

$$\sum_j b_{jk} \leq B_k \qquad (4.2)$$

Let us now assume that the traffic on the network is not homogeneous, but diverse and heterogeneous. All active traffic in the network can then be classified into I traffic classes. Furthermore, each traffic class i has a utility function $u_i(b)$ defined. If we assume that utility functions are additive, the overall utility generated in the network at any time is

$$U = \sum_{i=1}^{I} \sum_{j \in i} \sum_k u_i(b_{jk}) \qquad (4.3)$$

The optimal bandwidth allocation on such a model calculates the bandwidth b_{jk} to be allocated to flows on any link k belonging to any traffic class i in order to maximize the overall utility U generated in the network. This presents a multi-constrained optimization problem [LEB], which heavily depends on the defined utility functions.

During our research, we have analyzed the possible implementation of the bandwidth partitioning concept into this problem. A bandwidth partitioning scheme decouples the link bandwidth into I independent 'sublinks', where each of the sublinks serves one traffic class.

The partitioning is defined with the set of partitioning parameters $\alpha_{ik}, i = 1,...I,\ k = 1,...,K$. Each individual sublink i occupies the bandwidth $B_{ik} = \alpha_{ik} B_k$ on the link k. Sublinks can be treated independently; therefore the performance measures of interest can be easily calculated.

In order to analyse this idea in further detail, let us consider a simplified model which consists of a single link of bandwidth B. For this single link case, we can consider that bandwidth within each of the sublinks, $B_i = \alpha_i B$, is equally shared between n_i active traffic flows,

$$b_i = \frac{B_i}{n_i} = \frac{\alpha_i B}{n_i} \qquad (4.4)$$

Therefore, a traffic class is characterised by its transmission rate b_i and by its utility function $u_i(b_i)$. The objective of a bandwidth allocation scheme is to calculate the bandwidth for each traffic class that maximises the sum of the utilities over b_i subject to the link bandwidth constraints. The level of bandwidth allocated to each traffic flow depends heavily on the partitioning parameters. This optimisation problem can then be mapped on the problem of finding the vector of parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_I)$ which maximises the overall utility.

This is exactly what the DBP scheme was designed to achieve. In the scheme, the partitioning parameters α_i are not fixed and pre-defined, but are *variable*, and the parameters change with the change in the current end users' utility. The idea for the dynamic change in partitioning parameters in the DBP scheme is simple: every time the utility decreases for a certain value, a change in the partitioning parameters happens, in the direction which increases the overall utility.

In order to assess the current level of the generated utility, we define the *weighted normalised utility* Ψ, similar to [GAR00]:

$$\Psi = \frac{\sum \omega_i n_i u_i(b_i)}{\sum \omega_i n_i} \qquad (4.5)$$

The weighted normalised utility is used instead of the current utility level mostly because it shows the changes in the average utility more accurately then the overall utility. The overall utility level $U = \sum \omega_i n_i u_i(b_i)$ depends more on the number of active users n_i then on the current utility level $u_i(b_i)$.

ω_i are the scaling factors. The scaling factors are introduced to show that the defined traffic classes should not be treated with the same priority. Without the scaling factors, the generated utility will be determined by the number of the active flows from each of the classes. However, we argue that prioritisation is necessary in the multi-class Internet environment. It is far more complicated to serve a customer that uses a sophisticated video connection, than the one doing a simple file transfer. Furthermore, we must assume that a user of a video-conference is ready to pay more for the service than the user of file transfer.

The partitioning algorithm can now be defined as follows. If the normalised utility Ψ between two time instants, $t-\tau$ and t, decreases by more than a certain specified amount δ, a change of the partitioning parameters takes place

(1) *If* $\Psi(t-\tau)-\Psi(t)>\delta$ then the change will occur, $\xi=1$;

Else there will not be any change, $\xi=0$;

Let us introduce an indicator for the direction of change, $\theta_i(t)\in\{0,1\}$, where $\theta_i(t)=1$ indicates an increase, and $\theta_i(t)=0$ indicates a decrease. The direction of the change is defined by the change of the utilities for each traffic class,

$$\Delta u_i(t)=u_i(t)-u_i(t-\tau) \tag{4.6}$$

(2) *For each of the traffic classes* i, $i=1,\ldots,N$, *there is an indicator for change,* $\theta_i(t)$.

Determine the maximal individual decrease, $i_{\max}$, $\Delta u_{i\max}(t)\geq\Delta u_j(t)$, $\forall j$, $j=1,\ldots,I$

Then $\theta_{i\max}=1$*; for all other* i, $\theta_i=0$

The parameter with the largest decrease in the utility is increased, while all other parameters are decreased. Now a simple linear control algorithm is used for the update of partitioning parameters.

(3) *The new value for all partitioning parameters is calculated from:*

$$\alpha_i(t)=\alpha_i(t-\tau)+\xi\left[\varepsilon_{inc}\theta_i(t)-\varepsilon_{dec}(1-\theta_i(t))\right] \tag{4.7}$$

Parameters ε_{inc} and ε_{dec} are the increase/decrease parameters. The values for these parameters are not independent, since the sum of partitioning parameters needs to remain constant. In other words, the increase in one of the partitioning parameters needs to be followed by the simultaneous decrease in the value of other partitioning parameters. In the next section we will see that there are limitations when it comes to the decrease of the partitioning parameters. Therefore, if there are m traffic classes that are able to decrease the partitioning parameters, and the value for the decrease parameter is $\varepsilon_{dec} = \varepsilon$, the increase parameter is $\varepsilon_{inc} = m\varepsilon$. The value of the parameter ε is important, since this parameter defines the dynamics of the partitioning algorithm. Chapter 6 presents experiments with different values for the parameter ε, and shows the impact of this parameter to the overall network performance.

The linear control has been analysed in detail in section 3.3.2. If we compare the data from table 3.1 it is clear that the *additive increase, additive decrease* linear control is being used in the partitioning algorithm. The choice of linear control is a very interesting issue.

Important work by Chiu and Jain [CHI89] and Le Boudec [LEB] explains in more detail why is the choice of a additive increase, *multiplicative* decrease linear control mechanism optimal for achieving max-min fairness, i.e. for allocating bandwidth to individual flows so that the bandwidths are as equal as possible.

Further analytical work is necessary to prove which linear control mechanism is optimal for achieving utility-based fairness, especially in the case when multiple utility functions are used. In this Thesis we propose the use of additive increase, additive decrease control mechanism because of its simplicity, and show through the simulation that by using the scheme presented above, it is possible to increase the overall network performance.

4.4. Implementation Issues

4.4.1. Overview

The need for a simple implementation solution has been considered in the choice of the partitioning algorithm. Additive increase, additive decrease linear control mechanism is simple to implement in the TCP/IP network, especially since TCP congestion control uses similar linear control mechanism. The details of this implementation were presented in section 3.3.3. Furthermore, the implementation of the Dynamic Bandwidth Partitioning scheme is feasible in the MPLS-enabled network, and therefore in the MPLS-based Virtual Private Networks. The details of these implementations will be presented in this section.

However, there are several issues that are important to be discussed firstly. These issues present the extensions to the partitioning algorithm. They deal with the extreme situations, analysing the minimal partitioning parameters and the manual partitioning updates in the environment of very high rejection rates.

Minimal partitioning parameters are necessary for the network which is capable to provide strict bandwidth guarantees. Section 4.4.2 presents the problem in detail and introduces decrease permits for the partitioning algorithm.

The partitioning algorithm does not take under consideration the rejected traffic flows, i.e. the traffic flows that are not accepted in the network at the admission control point. By introducing manual partitioning updates after the rejection rate reaches a certain point, the algorithm performance greatly improves. The introduction of manual partitioning updates is described in section 4.4.3.

4.4.2. Analysis of Minimal Partitioning Parameters

The partitioning algorithm operates in a way that, when there is a need for the partitioning update, the partitioning parameter for the traffic class most affected increases, while the other parameters decrease. However, there are circumstances when this effect must be prevented. One of the key issues in the DBP scheme is the ability of the scheme to *guarantee* specific bandwidth levels for the traffic flows that require such guarantees, which was explained in section 4.3.2.

If we use an algorithm which decreases the partitioning parameters (and hence the overall level of bandwidth) without any control, there is nothing to prevent a sudden traffic burst from upsetting the guaranteed bandwidth levels.

Therefore, a mechanism is needed to prevent such an occurrence.

Firstly, let us analyse the bandwidth requirements. They can be defined through *minimal partitioning parameters*. The minimal partitioning parameters correspond to the amount of reserved bandwidth for the active traffic flows. More formally, if there is n_i active traffic flows belonging to the traffic class i, then minimal partitioning parameter $\alpha_{i\min}$ for this traffic class is

$$\alpha_{i\min} = \frac{n_i b_{i\min}}{B} \qquad (4.8)$$

where $b_{i\min}$ is the required minimal bandwidth level for traffic flows of traffic class i, and B is the link bandwidth.

The protection of bandwidth requirements is simply achieved by introducing a *Decrease Permit* for each traffic class, where

$$Decrease\_Permit[i] = (\alpha_i - \varepsilon > \alpha_{i\min}) \qquad (4.9)$$

The decrease of the partitioning parameter α_i is now possible only if $Decrease\_Permit[i] = 1$. These partitioning parameters will decrease for the decrease parameter ε. For all traffic classes with decrease permit equal to 0, the decrease of the partitioning parameter will not be performed.

The number of decrease permits defines the *increase* parameter for the traffic class with the largest individual utility change. If there are m traffic classes with decrease permits equal to 1, the partitioning parameter for the traffic class with the largest utility change will increase for $m\varepsilon$.

4.4.3. Manual Partitioning Update due to Blocking

The partitioning algorithm does not take into account traffic flows that are rejected at admission control. This means that if there is a large traffic load in one traffic class, the number of rejected flows belonging to another class will increase, and this phenomenon will have no influence on the normalised utility and therefore no influence on the partitioning algorithm.

There are two ways to overcome this problem. One is to somehow include the rejected traffic flows in the partitioning algorithm, and the other one is to introduce asynchronous, *manual* partitioning update for the case when the number of rejected traffic flows increases over some pre-defined limit.

The partitioning algorithm as presented here makes decisions on the basis of the *current* situation on the network link. It takes into account only the *active* traffic flows on the link. Therefore, it would be complicated to include the rejected flows in the partitioning algorithm without severely changing the algorithm itself.

The second way is to introduce manual partitioning updates. This introduction could go in the following two steps:

1. If the flow from the traffic class i is rejected, and if the rejection rate for that traffic class is above some pre-defined value of δ_{rej}^{i}, for example $\delta_{rej}^{i} = 0.02$, then the manual change of partitioning is done
2. The manual change in partitioning is done in the same way as before, where the traffic class in question increases its partitioning parameter, and the other traffic classes decrease their partitioning parameters

With this additional rule in the partitioning algorithm, another level of satisfying the Service Level Agreement with the users is achieved. The network is now able to guarantee not only the bandwidth levels, but also the blocking probability.

4.4.4. Implementation in the MPLS-enabled Networks

This section explains a mechanism for the implementation of the Dynamic Bandwidth Partitioning scheme in MPLS-enabled networks. This implementation has been studied in more detail in one of our publications [RAK3].

The main features of Multiprotocol Label Switching (MPLS) have been discussed already in Chapter 2. Here we study some additional MPLS features and explain the way Dynamic Bandwidth Partitioning can be used in the MPLS-enabled network to increase the network efficiency.

The most important feature of MPLS is the establishment of Constrained-Routed Label Switched Paths (CR-LSPs) for traffic aggregates, and enabling the control for the bandwidth allocation on individual links in CR-LSPs. If we assume that the traffic from individual traffic classes is aggregated on the link and transported via CR-LSP, it is possible to make a simple analogy between these CR-LSPs and "bandwidth partitions" which have been presented throughout this Chapter. After that, a method of dynamic re-negotiation and resizing of CR-LSPs is needed for the implementation of the dynamic partitioning algorithm. This section presents briefly the mechanisms for establishing CR-LSPs, and the proposed mechanism for their dynamic resizing.

Before moving to further analysis, it is necessary to define several MPLS features.

Constrained-Routed Label Distribution Protocol (CR-LDP) [JAM01] is a set of extensions to Label Distribution Protocol [RFC3036] specifically designed to facilitate constraint-based routing of LSPs. Like LDP, CR-LDP uses TCP sessions between Label Switched Router (LSR) peers and sends label distribution messages along the sessions.

The basic flow for LSP set-up using CR-LDP is shown in the figure 4.1 below. The ingress LSR, LSR A, determines that it needs to set up a new LSP to LSR C. It determines that the route for the new LSP should go through LSR B. LSR A builds a LABEL_REQUEST message with an explicit route (B, C) and the details of the traffic parameters requested for the new route. LSR A reserves the resources it needs for the new LSP, and then forwards the LABEL_REQUEST message to LSR B on the TCP session.

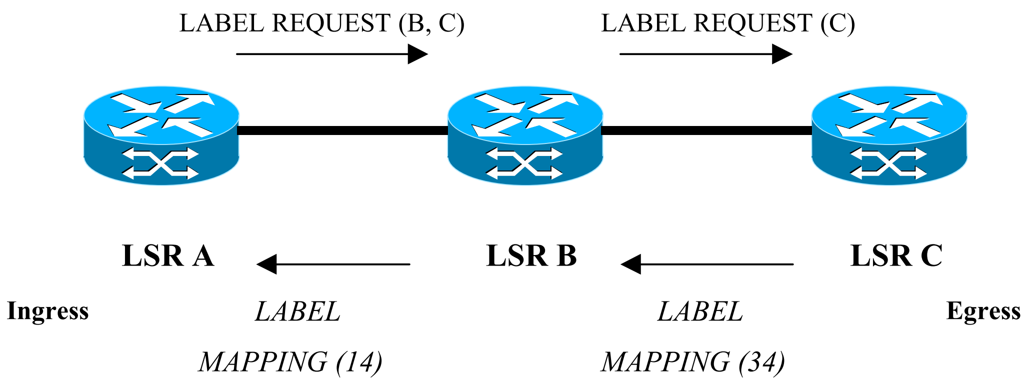

Figure 4.1. CR-LDP LSP Set-up Flow

When the LABEL_REQUEST message from LSR B comes to LSR C, it determines that it is the egress router for the new LSP. It allocates a label (e.g. 32) to the new LSP and distributes the label to LSR B in a LABEL_MAPPING message, which contains details of the final traffic parameters reserved for the LSP. The exact definition of LABEL_REQUEST and LABEL_MAPPING messages can be found in [RFC3036]. LSR B receives the LABEL_MAPPING message and matches it to the original request using the LSPID TLV. LSPID TLV is a unique identifier of a LSP, and it is exactly defined in [JAM01]. LSR B then finalises the reservation, allocates the label (14) for the LSP, sets up the forwarding table entry, and passes the new label to LSR A in a LABEL_MAPPING message.

Understanding of this basic principle of LSP set-up and label distribution in MPLS is important for the implementation of the Dynamic Bandwidth Partitioning scheme. We assume that the traffic aggregates are forwarded through the network via the LSPs. The sizes of LSPs are then defined by the partitioning parameters.

Label Distribution Protocol uses a *Type/Length/Value* (TLV) encoding scheme to encode much of the information carried in LDP messages. For example, there is a Generic Label TLV, an ATM Label TLV, or a Frame Relay TLV. TLV is encoded as a 2 octet field that uses 14 bits to specify a Type and 2 bits to specify behaviour when an Label Switch Router does not recognise the Type, followed by a 2 octet Length field, followed by a variable length Value field (see figure 4.2).

U bit is **unknown** TLV bit. Upon receipt of an unknown TLV, if U is clear (U=0), a notification must be returned to the message originator; if U is set, the unknown TLV is silently ignored. *F bit* is **forward unknown** TLV bit. This bit applies only when the U bit is set and the LDP message containing the unknown TLV is to be forwarded. If F is clear, the unknown TLV is not forwarded with the containing message; if F bit is set, the unknown TLV is forwarded with the containing message.

0	1	15	31
U	F	Type	Length
Value			

Figure 4.2 Type/Length/Value Field

Type field encodes how the Value field is to be interpreted. Three TLVs given as an example above have type field values defined as 0x0200, 0x0201, and 0x0202, respectively. *Length field* specifies the length of the Value field in octets. *Value field* is an octet string of Length octets that encodes the information to be interpreted as specified by the Type field.
As an example, let us take a look at the ATM Label TLV (figure 4.3).

		15	31
0	0	ATM Label (0x0201)	Length
Res	V	VPI	VCI

Figure 4.3. ATM Label TLV

Res field is reserved, it must be set to zero and must be ignored. *V-bits* is a two bit switching indicator. It defines the significancy of both VPI and VCI. For example, if it is 00, both VPI and VCI are significant. VPI and VCI are typical identifiers in the ATM architecture, *VPI* is the Virtual Path Identifier, and *VCI* is the Virtual Circuit Identifier [ONV94].

For successful implementation of Dynamic Bandwidth Partitioning in the MPLS network, it is necessary to enable dynamic modification of the bandwidth reserved for the CR-LSP. The procedure for that is defined in [ASH01]. Resource modification is defined only when the LSP is already set up, and it can be requested only from the ingress LSR. To modify the characteristics of a LSP, the ingress router sends a LABEL_REQUEST message. In that message, the *Traffic Parameter TLV* will have the new requested value. The Traffic Parameter TLV carries the information about the peak data rate and the peak burst size of the traffic that is using CR-LSP. The exact definition of the Traffic Parameter TLV is given in [JAM01]. The route, traffic class or set up priority for the LSP can also be changed, but that mechanism is beyond the scope of our problem.

When an intermediate LSR receives the LABEL_REQUEST message, after checking the LSPID TLV, it identifies that it already serves that particular LSP. Then it takes the newly received Traffic Parameter TLV and computes the new bandwidth. In this computation, the LSR should be aware of the value of the partitioning parameter of the link LSP is using. Compared with the already reserved bandwidth for the LSP, LSR now reserves only the difference in the bandwidth requirements. This prevents the LSR from doing double booking of bandwidth.

Each originating router monitors bandwidth use on each CR-LSP, and determines when allocated bandwidth needs to be increased or decreased. This information comes from the change in the partitioning parameters on the links belonging to the CR-LSP and to the link-state of those links. In making a bandwidth modification, the originating router determines the QoS resource management parameters, including bandwidth-in-use and bandwidth allocation threshold (partitioning parameter).

In summary, additional information necessary for the implementation of the Dynamic Bandwidth Partitioning scheme in MPLS includes:

- Each router should be informed about the value of the partitioning parameter of all the links that are connecting it to other routers. This information can come from a centralised controller, or it can be distributed in the network.
- Every change in the partitioning parameters should trigger the modification of CR-LSPs that use the link which has changed its partitioning. The modification can be requested only from the ingress router for the CR-LSP.

It is interesting to notice that a very similar problem was analysed in an earlier Internet Draft [ASH99]. In that work, available bandwidth is allocated in discrete changes to each of the three virtual networks corresponding to high-priority key services, normal priority services and best-effort services. In [ASH99], bandwidth changes in virtual network bandwidth are determined by edge switch/routers based on an overall aggregated bandwidth demand for bandwidth (not on per-connection demand basis). Based on the aggregated bandwidth demand, the edge switch/routers make periodic discrete changes in bandwidth allocation, that is, either increase or decrease bandwidth on the CR-LSPs constituting the virtual network bandwidth. The analogy of this approach with our DBP scheme is obvious.

Therefore, MPLS standards already provide a basis for successful implementation of the Dynamic Bandwidth Partitioning scheme. CR-LSP modification is possible and fairly simple, as defined in [ASH01]. Possible problems may arise from the fact that there is additional signalling information that needs to be distributed within the MPLS network. The added complexity needs to be evaluated. The second problem is the question of whether MPLS network is fast enough to react on the constant changes in the link partitioning. That is why small values for the decrease parameter should be chosen.

4.4.5. Implementation in the VPN Environment – Hierarchical DBP

This section analyses another very important implementation of the Dynamic Bandwidth Partitioning scheme. It has already been stated in this Thesis that one of the situations where bandwidth partitioning is necessary is when there is a need to lease the network capacity to large customers who are prepared to pay for it. A Virtual Private Network (VPN) is a private network constructed within a public network infrastructure. In recent years, especially with the growing interest in electronic commerce, VPNs have experienced a dramatic growth.

Virtual Private Networks can be categorised into three types:

- *Remote-access* VPNs – to connect remote mobile users to the corporate LAN
- *Intranet* VPNs – to connect remote branch offices, and to connect branch offices to the main office of a company
- *Extranet* VPNs – to give business partners limited access to the corporate LAN

From the point of view of bandwidth allocation, the second type of VPN is the most interesting. An *Intranet* VPN is used for communication between remote corporate sites. Users on those sites have a feeling that they all belong to one secure and fast local network. The users should be able to freely communicate between the sites, sending and receiving all types of Internet traffic.

The design problem for the point of bandwidth management is how to allocate the bandwidth of a core Internet link which is used by a number of such Intranet VPNs and by the 'normal' Internet traffic. The bandwidth allocation should be optimal, which means it should bring high network utilisation, at the same time satisfying the performance requirements of all users. On the other hand, the allocation should be simple and transparent to the users, since the users should not care about the exact way the network capacity is shared. The users have their expectations and are ready to pay for the service.

The majority of the theoretical approaches to the problem [GAR00][DUF99] [GOY99][MIT97], which were studied in more detail in section 3.4.5, assume a network as a resource being shared by a number of 'customers', where each of the 'customers' represents a VPN. From the point of view of bandwidth management, a key feature of the VPN environment is to optimise the bandwidth sharing on two levels. One level is sharing the bandwidth between VPNs, and the other is sharing the bandwidth *within* each of the VPNs. Complete satisfaction for the customers can come if the network utilisation is very low, or if the traffic belonging to different VPNs is fully isolated, i.e. if the network is *fully partitioned*. All of the concepts discussed in section 3.4.5 have in common the necessity for hierarchical traffic differentiation, and partitioning of the link capacity between active VPNs, where that partitioning can be virtual (at the admission point), or physical (by setting the scheduling parameters in multi-priority queues in the intermediate routers).

In [RAK2] we have investigated possible implementation of the Dynamic Bandwidth Partitioning scheme into the bandwidth management in the VPN environment. Since bandwidth partitioning is the natural solution for bandwidth management for VPNs, we have proposed the Dynamic Bandwidth Partitioning for increasing the level of the overall network performance.

As explained before, the bandwidth allocation problem for VPNs is on two levels. Therefore, the DBP scheme needs to be extended to *Hierarchical Dynamic Bandwidth Partitioning*. In this scheme, the link is partitioned not between traffic classes, but between *traffic subjects*, where a traffic subject can be either a traffic class or a VPN, which serves a number of traffic classes. We can define partitioning parameters A_i and α_{ij}, where A_i refers to the VPN, and α_{ij} refers to the j^{th} traffic class within the i^{th} VPN. Now VPN i has A_iB bandwidth available on the link, while traffic class j within the VPN i has $\alpha_{ij}A_iB$ bandwidth available. The utility U_i for the individual VPN, and the overall utility U can now be expressed in the following way:

$$U_i = \sum_j \omega_{ij} n_{ij} u_{ij}\left(b_{ij}\right) \qquad (4.10)$$

$$U = \sum_i \Omega_i n_i U_i \qquad (4.11)$$

Where U_i is the utility generated on the i^{th} VPN, Ω_i are the scaling parameters, and $n_i = \sum_j n_{ij}$. Now, whenever the weighted normalised utility $\Psi = U \big/ \sum n_i \Omega_i$ decreases by a certain amount the resizing happens by changing the partitioning parameters A_i. Furthermore, the adaptation of the partitioning parameters α_{ij} within each VPN proceeds as explained above. This means that there is a *continuous dynamic allocation* both within each individual VPN and on the link as a whole, maximising the overall utility.

Resizing based on the change for the normalised utility ensures that in the case that any of the VPNs suffers a sudden burst of traffic, the normalised utility on that VPN will decrease, since the bandwidth levels for all active traffic flows belonging to that VPN will decrease. The partitioning algorithm is designed to reduce the impact of such an event, by increasing the capacity allocated for the VPN that suffered the traffic burst. The introduction of scaling parameters ω_i ensures that the bursts in the high priority traffic have greater influence than the burst in the low priority traffic.

4.5. Summary

This Chapter defined the main contribution of the Thesis, the Dynamic Bandwidth Partitioning scheme. The first section gave an introduction to the problem of utility-based resource allocation. The main point is that the available resources need to be shared so that the mean utility experienced by end-users is maximised. An overview of the related work on the end user-based optimisation of resources was given. Each of the analysed works contributed in developing the new scheme.

However, the Dynamic Bandwidth Partitioning scheme presents a novel approach. The scheme is based on the idea that the available bandwidth should be partitioned among the defined traffic classes, so that the forwarding of each of the traffic classes is independent. Furthermore, the dynamic modification of the partitioning is introduced through a simple linear control 'additive increase, additive decrease' algorithm. The scheme uses the information about the fluctuation in the *normalised utility* to make the change in the way the link bandwidth is partitioned.

The scheme was presented through its most important features: admission control, bandwidth allocation and partitioning algorithm.

The implementation issues have been studied carefully and they have contributed to the final design of the scheme. Section 4.4 studies the implementation issues, starting with simple extensions to the partitioning algorithm. Very important is the possibility of implementing the DBP scheme in an MPLS-capable network, where dynamic partitioning updates can be performed by dynamic resizing of the established LSPs. A number of IETF standards explain the way this can be done.

Finally, this Chapter presented the extension of the scheme to Hierarchical Dynamic Bandwidth Partitioning, which can prove to be an efficient solution for the bandwidth management for the IP environment of Virtual Private Networks.

The next Chapter presents the simulation model that has been used in the evaluation of the scheme and the main features of the developed simulator.

CHAPTER 5

SIMULATION MODELLING

5.1. Introduction

There are two ways of measuring the performance characteristics of a new resource allocation scheme: through mathematical analysis or through a simulation model. In the case of the Dynamic Bandwidth Partitioning scheme, mathematical analysis includes solving optimisation problems with multiple constraints. To analyse mathematically the design of the optimal resource allocation scheme in a multi-class network is a very complicated problem. The level of complexity rises rapidly with the increase in the number of different traffic classes.

To be able to make measurements using different utility functions and adaptive traffic differentiation, and to experiment with different parameters in the Partitioning algorithm, simulation models were used.

A specially built event-driven simulator has been designed in the C programming language. By using steady-state simulation, ignoring the initial state of the simulated system, we have gathered sufficient data which enabled us to study the performance of the scheme in detail, and to discover its properties.

This Chapter will describe the details of the simulation model, including the analysis of the network and traffic model. The traffic differentiation is described in detail, including the definition of the utility functions and the optimal parameters for the utility functions. Finally, section 5.4 presents the simulator in detail and provides the mathematical analysis which is used for the validation of the simulation.

5.2.Network Model

The network topology that we have used in our simulation is simple, consisting of a single link with three traffic sources generating the traffic that belongs to three different traffic classes (see figure 5.1).

Our previous work [RAK5][RAK7] included the study of a multi-link network topology, in which the utility for the real-time traffic was dependent on the number of hops the traffic traversed. For the purpose of that research different 4-node and 6-node arbitrary topologies were used.

However, the objective of the Thesis is the full definition of the Dynamic Bandwidth Partitioning scheme. For this purpose a single link network model is used, as the simplest model that provides space for the detailed analysis of the scheme. In the research presented in this Thesis, the utility is calculated on the basis of the allocated bandwidth, without considering the number of hops. Further research will describe the optimal implementation of this concept in large network topologies. Possible directions for future work are described in detail in Chapter 7.

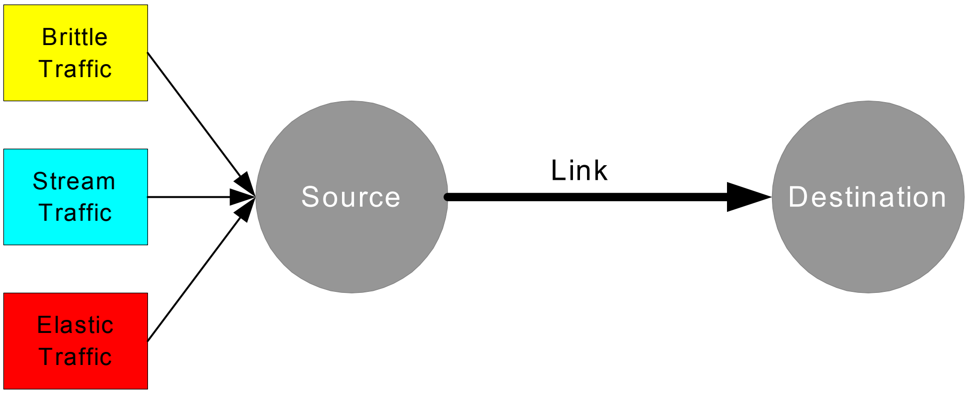

Figure 5.1. Single Link Model

5.3. Traffic Model and Utility Functions

The Dynamic Bandwidth Partitioning scheme is based on the notion of *utility*. In this Section we explain the traffic differentiation used for the simulation experiments, define the utility functions for each of the traffic classes, and in that way prepare the ground for defining the *connection utility*, which is the main performance parameter measured in the simulation model.

In the network model used in the simulation, we use a traffic differentiation where entire Internet traffic is differentiated into three major traffic classes:

1. **Brittle traffic**
2. **Stream traffic**
3. **Elastic traffic**

Each of these three traffic classes is discussed in more detail in the following sections. There is a utility function defined for each of the traffic classes, and the analysis that follows will show that the traffic differentiation defined here can be considered as a basic traffic differentiation within which any new traffic class can be defined. We argue that this is true because, by using the three utility functions defined for these three traffic classes, it is possible to define almost any new utility function.

5.3.1. Brittle Traffic

Brittle (hard real-time) traffic requires strict end-to-end performance guarantees and does not show any adaptive properties. A brittle traffic flow is not allowed to enter the network if there is not enough bandwidth available. While in the network, a brittle traffic flow occupies a constant amount of bandwidth on the link. Analogies for this traffic class can be found in the CBR traffic class of ATM, Guaranteed Service in the Integrated Services architecture, or Expedited Forwarding per-hop behaviour in the Differentiated Services architecture. Typical applications belonging to this traffic class would be video telephony, highly secure data transactions, telemedicine etc. The user of a brittle traffic connection is interested only in high level of quality of service. If the network is unable to guarantee the required performance for a traffic flow belonging to this traffic class, the end-users' utility will be 0. That is why for the brittle traffic class we use a very simple utility function (Fig.5.2):

$$u_b(b_b) = \begin{cases} 1, \ \mathit{if} \ b_b \geq b_{b\min} \\ 0, \mathit{if} \ b_b < b_{b\min} \end{cases} \qquad (5.1)$$

where b_b is the allocated bandwidth, and $b_{b\min}$ is the minimum required bandwidth.

It is interesting to note at this point that similar utility functions have been used in the literature. For example, Sarkar and Tassiulas [SAR99] use a stepwise utility function for video applications. In their analysis, each added layer of bandwidth to the video flow increases the utility by a certain fixed value. In our analysis a simplified version of their utility function is used, with only one non-zero utility level.

When it comes to the level of *priority* brittle traffic should receive in the scheduling and resource allocation mechanisms, there are several reasons why brittle traffic needs to be considered as the traffic class of the highest priority.

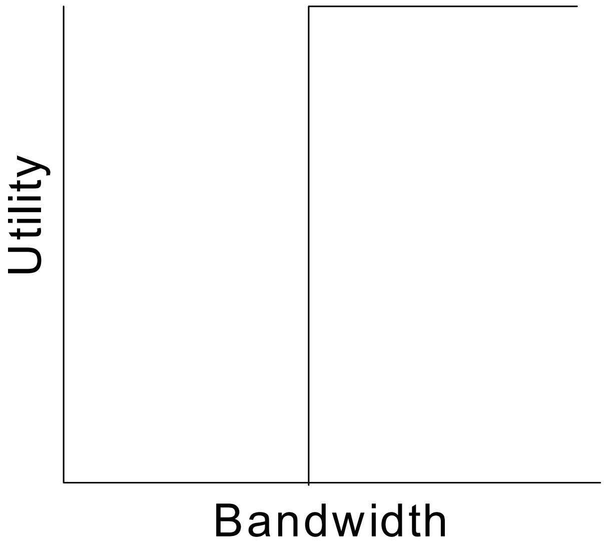

Figure 5.2. Utility Function for Brittle Traffic

Firstly, because of its strict bandwidth and delay requirements, brittle traffic flows cannot depend on the randomness of the network congestion. If a brittle traffic flow suffers from the network congestion in a way that its end-to-end delay increases, or that the bandwidth level is smaller then required, the end-users of the brittle traffic connection will almost immediately refuse to use the network. Therefore, the choice is simple; either enable the network to guarantee the required performance, or expect that the end-users will change the network provider.

This is where the second reason for giving high priority lies: the loss of revenue. Although the pricing system in the Internet is still uncertain, it is widely assumed that applications such as video conferencing, video telephony, video and TV on demand, and other belonging to the brittle traffic class will be priced higher than the stream or elastic traffic classes. Therefore, in order to achieve high revenues expected, it is necessary to assign high priority to the brittle traffic class.

5.3.2. Stream Traffic

Stream (adaptive real-time) traffic results from audio and video applications and requires the network to provide a *minimum* level of network performance guarantees. While in the network, adaptive traffic connections can adapt to the amount of resources the network allocates them.

For example, stored video and audio sequences accessed remotely across the network can be considered as stream traffic. These applications have become very popular within the confines of the WWW, contributing now a large percentage of overall Internet traffic. Delay and loss are much less critical under these circumstances so that throughput could be identified as the dominating QoS factor governing non-real-time multimedia communications. Playout buffers and retransmissions serve well those types of applications.

The adaptation of audio and video applications can be achieved at several layers of the network protocol stack. At the data link layer, error control and adaptive reservation techniques can be used to protect against variation in error and available rate. At the transport layer, dynamic re-negotiation of connection parameters can be used for adaptation. At the application layer, the application can adapt to changes in network conditions using several techniques, including different compression algorithms, layered encoding, adaptive error control, bandwidth smoothing, etc. An excellent survey of application layer techniques that can be used to enhance the adaptability of applications can be found in [VAN99]. The details of these techniques are beyond the scope of this Thesis. However, it is essential to understand that it is possible, due to remarkable research achievements, to abandon the circuit-switched concept for transporting video and audio traffic, and guarantee the appropriate network performance to the audio and video traffic by guaranteeing a certain *minimal* bandwidth level.

For the stream traffic, traffic flows require some minimum level of bandwidth $b_{s\,min}$. Traffic belonging to this traffic class is rate-adaptive. It is able to change its rate between the minimum required rate $b_{s\,min}$, and the peak rate $b_{s\,max}$. Nevertheless, admission control for this traffic class is necessary, and therefore the optimal number of active traffic flows on the link should be finite. If n stream traffic flows use a single link of bandwidth B, the optimal number of flows, n_{opt} that maximises the overall utility, $nu\left(B/n\right)$, must not be infinite. The utility function that models such behaviour is (figure 5.3):

$$u_s\left(b_s\right)=1-e^{-a_{s1}\frac{b_s^2}{a_{s2}+b_s}} \qquad (5.2)$$

where b_s is the allocated bandwidth. The function (5.2) is the modified form of the function used by Shenker and Breslau in [SHE98]. Other authors [CAO99] have used functions of similar shape to model the utility of rate-adaptive traffic connections. The most important feature of this function is its non-concavity.

It is interesting to make a connection between the utility function and the set of applications (playback video and audio) we are modelling by this utility function. The small levels of bandwidth are not very useful, so that at low bandwidths the marginal utility of additional bandwidth is small. At high bandwidths the signal quality is already good enough and the marginal utility of additional bandwidth at high bandwidths is also small. It is only at intermediate levels, that the marginal utility of extra bandwidth is significant. We can see from figure 5.3 that at low bandwidth values the function is convex.

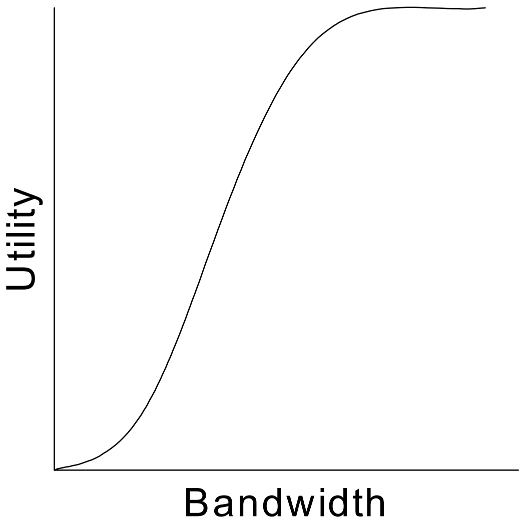

Figure 5.3. Utility Function for Stream Traffic

The utility function (5.2) has two shaping parameters, a_{s1} and a_{s2}. The choice of these two parameters is crucial for the definition of the function shape. The existence of the two parameters gives the network operators freedom in adapting the shape of the function to the applications that are being modelled.

The values for the parameters a_{s1} and a_{s2} depend on the value for minimal and maximal required bandwidth for the stream traffic, $b_{s\min}$ and $b_{s\max}$, and on the *shaping rule*. The shaping rule is a simple rule defining the steepness of the function. It is expressed through a simple relation:

$$u_s\left(b_s = 0.5b_{s\max}\right) = \theta_s \qquad (5.3)$$

This rule defines that the value of the function for the half of the maximal bandwidth is equal to some parameter θ_s, where $0 \le \theta_s \le 1$. This is also the first equation used for determination of the shaping parameters a_{s1} and a_{s2}. The second equation is more complicated. The minimal required bandwidth level $b_{s\min}$ is defined as the level of bandwidth after which the utility for the user of a stream traffic flow increases rapidly. Therefore, $b_s = b_{s\min}$ should be the point where the function (5.2) stops being convex, and starts being concave. It is well known that this happens at the point where the second-order derivative of the function is equal to zero.

Therefore,

$$\frac{\partial^2 u_s(b_s)}{\partial {b_s}^2} = 0 \text{, for } b_s = b_{s\min} \tag{5.4}$$

The equation (5.4) is the second equation necessary for the calculation of shaping parameters a_{s1} and a_{s2}.

From (5.3) we have

$$1 - e^{-\frac{a_{s1}{b_s}^2}{a_{s2}+b_s}} = \theta_s \text{, for } b_s = 0.5 b_{s\max} \tag{5.5}$$

It is easy to calculate from there

$$a_{s1} = \frac{4a_{s2} + 2b_{s\max}}{b_{s\max}^2} \ln\left(\frac{1}{1-\theta_s}\right) \tag{5.6}$$

For simplicity, let us define $\rho = \ln\left(\frac{1}{1-\theta_s}\right)$. Equation (5.6) is the first equation for the calculation of shaping parameters. The second equation is found after the second-order derivation of the function $u_s(b_s)$. This derivation is easy to find:

$$\frac{\partial^2 u_s(b_s)}{\partial {b_s}^2} = \left[a_{s2}^3 + \left(2b_s - 4a_{s1}b_s^2\right)a_{s2}^2 - 4a_{s1}b_s^3 a_{s2} - \frac{a_{s1}b_s^4}{2}\right]\frac{2e^{-\frac{a_{s1}b_s^2}{a_{s2}+b_s}}}{\left(a_{s2}+b_s\right)^4} \tag{5.7}$$

What we need to find are a_{s1} and a_{s2} for which the second-order derivation (5.7) is equal zero for $b_s = b_{s\min}$, and at the same time equation (5.6) is fulfilled for a defined θ_s. Hence, by knowing $b_{s\min}$, $b_{s\max}$ and θ_s (the shaping rule), we can calculate the shaping parameters.

It is clear that (5.7) is equal to zero when the polynomial in the brackets is equal to zero. Therefore, we will continue to analyse only the polynomial. After inserting the relation (5.6) into the polynomial, the final equation for the calculation of a_{s2} is derived:

$$\left(1-\frac{8b_{s\min}^{2}\rho}{b_{s\max}^{2}}\right)a_{s2}^{3}+\left(b_{s\min}-\frac{4b_{s\min}^{2}\rho}{b_{s\max}}-\frac{8b_{s\min}^{3}\rho}{b_{s\max}^{2}}\right)a_{s2}^{2}-\left(\frac{4b_{s\min}^{3}\rho}{b_{s\max}}+\frac{2b_{s\min}^{4}\rho}{b_{s\max}^{2}}\right)a_{s2}-\frac{b_{s\min}^{4}\rho}{b_{s\max}}=0 \quad (5.8)$$

Equation (5.8) presents a third-order polynomial which is easy to solve by using some of the simulation packages available (e.g. Matlab). The non-negative roots of this polynomial present the possible values for the shaping parameter a_{s2}. After that value is calculated, the shaping parameter a_{s1} is easily calculated from (5.6).

The reason why the shaping parameters are important is twofold. Firstly, the choice of these parameters gives the network operators the opportunity to *choose* the utility function to model the stream traffic. It will be shown later in this section that different values for a_{s1} and a_{s2} result in rather different shapes of the functions. Secondly, by showing that it is possible to change the utility function, we have achieved a generalisation of the problem that is being analysed in this Thesis. Although we are using only three broad traffic classes, the analysis in this section shows that the three traffic classes defined here give enough basis for the modelling of any other traffic class or any single user-application, however specific it is.

Table 5.1 presents the values for the shaping parameters for various bandwidth requirements and shaping rules.

Table 5.1. Shaping Parameters for the Stream Utility Function

$b_{s\min}$	$b_{s\max}$	θ_s	a_{s2}	a_{s1}
0.1	2	0.5	0.0254	0.7107
0.1	2	0.9	0.0612	2.4436
0.3	2	0.9	0.7436	4.0148
0.4	2	0.9	2.8464	8.8567

It is clear from the Table 5.1 that as $b_{s\min}$ increases a_{s2} also increases. The question then arises, what impact does that have on the shape of the utility function? Figure 5.4 shows a simple comparison of three utility functions with different a_{s2} parameters. It is obvious that as a_{s2} increases, the function becomes *'more convex'* for the low bandwidth levels. This is due to larger minimal required bandwidth $b_{s\min}$. The more minimal bandwidth a traffic flow requires, the more its utility function will be convex for the larger portion of the x-axes.

On the other hand, we can see form the Table 5.1 that larger value for θ_s results in the larger value for the shaping parameter a_{s1}. The result of that can be observed on Figure 5.5.

The larger value for a_{s1} makes the utility function steeper, and brings the shape of the function closer to the shape of the brittle utility function.

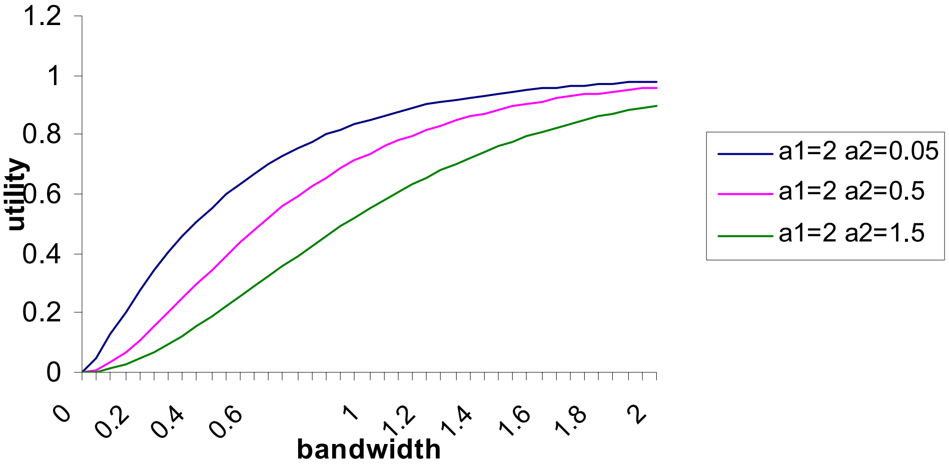

Figure 5.4. Impact of the Shaping Parameter a_{s2} on the Utility Function for Stream Traffic

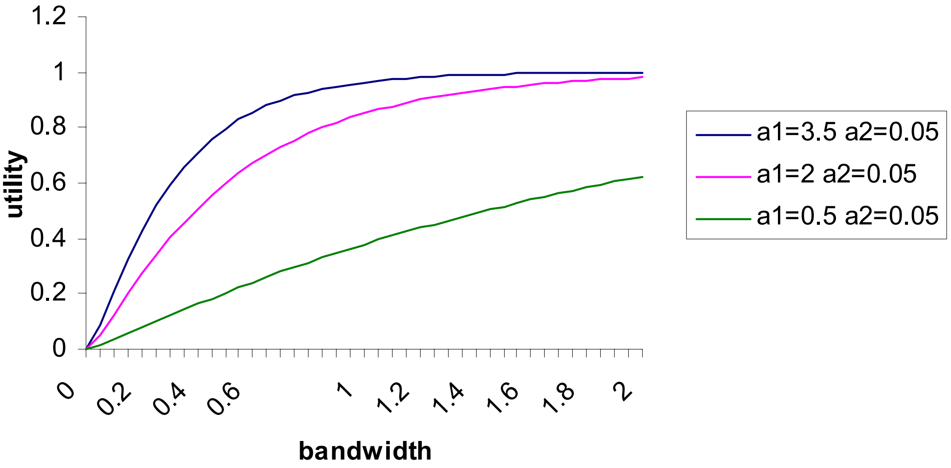

Figure 5.5. Impact of the Shaping Parameter a_{s1} on the Utility Function for Stream Traffic

This analysis showed how different bandwidth and performance requirements influence the utility functions. The impact of the shapes of utility functions on the network performance when Dynamic Bandwidth Partitioning scheme is used will be analysed in the next Chapter. Experimental results presented there will show how the performance of the partitioning algorithm is influenced by the shaping parameters.

5.2.3. Elastic Traffic

Elastic (best-effort) traffic flows are established for the transfer of digital documents (files, pictures, e-mail), and only have loose response time requirements. There is no admission control for this type of traffic.

In the case of elastic traffic flows, there is no minimum bandwidth requirement. There is no need for the admission control, and optimal number of active flows is infinite. The utility function that models such requirements should be always concave, but not linear. The following utility function is used for elastic traffic (figure 5.6):

$$u_e(b_e) = 1 - e^{-\frac{a_e b_e}{b_{e\,max}}} \qquad (5.9)$$

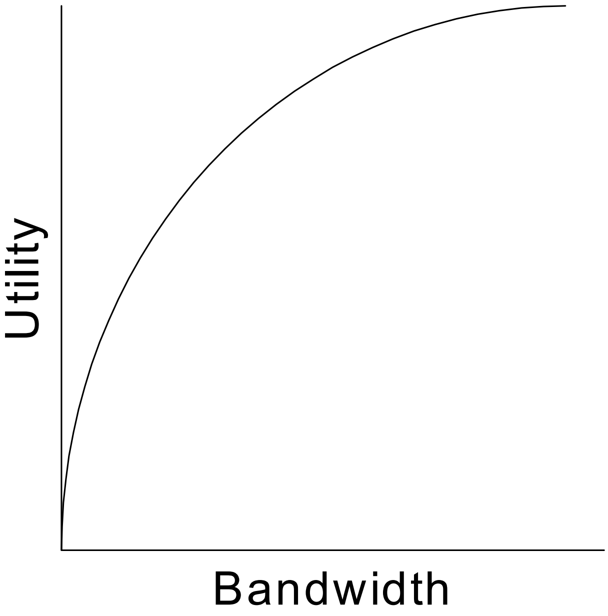

Figure 5.6. Utility Function for Elastic Traffic

b_e is the allocated bandwidth, and $b_{e\,max}$ denotes the peak rate for the elastic flow. We consider that the peak rate for elastic flows is equal to the full link capacity, $b_{e\,max} = B$. Following the same logic as in the previous section, the question is what is the optimal number of elastic flows n_{opt} that maximises the utility on the link, $nu_e\left(\frac{B}{n_e}\right)$. Since the utility function $u_e(b_e)$ is always concave, the optimal number of traffic flows on the link is $n_{opt} = \infty$, i.e. the optimal admission policy is "accept all". This admission policy is a feature of current best-effort communication network. Figure 5.6, shows that the marginal utility of extra bandwidth is larger when the bandwidth is small. In the area of high bandwidth, adding extra bandwidth does not improve utility as much as when bandwidth is small.

The shape of the function (5.9) depends on only one parameter, a_e. This parameter has a large impact on the shape of the utility function. Function $u_e(b_e)$ is concave, with the defined maximal bandwidth. Therefore, the calculation of the shaping parameter is simpler than for the stream utility function. Only a shaping rule is needed for the calculation:

$$u_e(b_e = 0.5 b_{e\max}) = \theta_e, \qquad 0 \le \theta_e \le 1 \qquad (5.10)$$

From there,

$$1 - e^{-\frac{0.5 a_e b_{e\max}}{b_{e\max}}} = \theta_e \qquad (5.11)$$

It is easy to calculate the shaping parameter:

$$a_e = 2\ln\frac{1}{1-\theta_e} \qquad (5.12)$$

Similar to the analysis in the previous section, it is interesting to see what impact does the shaping parameter a_e have on the utility function $u_e(b_e)$. Table 5.2 shows the values for a_e for three different shaping rules. We can see that, as θ_e increases, the corresponding a_e also increases. The impact of the increasing a_e on the shape of the utility function can be seen at Figure 5.7. The larger a_e is, the function will be steeper, and the utility value will approach 1 for smaller bandwidth levels.

Table 5.2. Shaping Parameter for the Elastic Utility Function

θ_e	0.5	0.75	0.9
a_e	1.3863	2.7726	4.6052

Similar to the previous section, it is possible to conclude at this point that elastic utility function (5.9) with the appropriate parameter a_e gives a good basis for modelling a number of different elastic applications. By choosing a_e to be larger, higher priority file transfer can be modelled, since in that case the marginal utility for low bandwidths is large, showing that it is very important for the application to be able to use as much bandwidth as possible. By choosing small a_e lower priority elastic applications can be modelled.

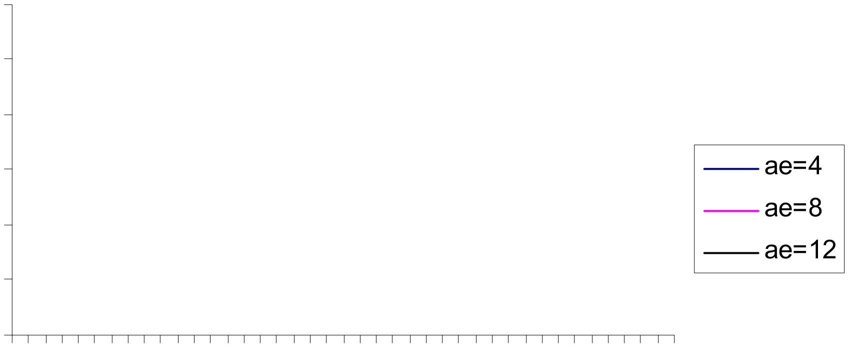

Figure 5.7. Impact of the shaping parameter a_e on the Utility Function for Elastic Traffic

Nevertheless, the overall priority elastic traffic should obtain in the integrated network which serves all three traffic classes defined here, should be low. We use elastic traffic class to model the application which are not critically sensitive on the bandwidth (and delay) levels obtained in the network.

5.4. Simulator

This section describes the simulator in more detail. The specially developed simulator is an event-driven simulator, so firstly discrete-event simulation is discussed. This is followed by a brief analysis of the simulator. The simulation results are useless if the simulation is not previously validated. Simulation validation consists of comparing the simulation results with a theoretical model. Detailed validation of the simulator used for our experiments is given in section 5.4.3.

5.4.1. Discrete-Event Simulation

The main features of a discrete-event approach [PIT93] are that the components of an actual network are represented within software and the events that would occur in the operation of the real network are represented by the operation of the software. The function of the simulator is to generate the traffic events, calculate the response of the network elements to these events, and to record and analyse the relevant data measures.

The fundamental elements of a discrete-event simulation are the events; these may change the state of the system, or they may schedule an action to do with measurement, monitoring, or the progress of the simulation [PIT93][LAW91]. A mechanism for keeping track of the simulation time is required, and there are two alternatives for advanced simulated time: *next event time advance* and fixed increment time advance. In general, fixed increment time advance suffers either a lack of timing accuracy if the increment is too large, or a waste of computing effort if the increment is too small. Thus the former mechanism, which combines timing accuracy and computing efficiency, is the normal approach for most cases. The simulator that is developed for our research is also event-driven, i.e. uses next event time advance mechanism.

There are a number of standard functional components, either variables or procedures, which together form the basic organisation of a discrete-event simulator [PIT93][LAW91]. The *system state* consists of the state variables which describe the system at a particular instant in time. The current value of simulation time is stored in a variable called the *simulation clock*. The *event list* is a time ordered list used to store forthcoming events. *Statistical counters* are variables which store statistical information about the system performance. Initialisation and configuration of the system model is performed by the *initialisation routine*. The *timing routine* gets the next event from the event list and advances the simulation clock to the time at which that event occurs. *Event routines* modify the system state when a particular event occurs. The *report generator* calculates performance estimates from the statistical counters and saves the information to file. Finally, the *main program* organises the overall progression of the simulation, its primary job being to call the timing routine and the appropriate event routine (for the next event just extracted from the event list) until the simulation has progressed to its scheduled stopping time.

5.4.2. Analysis of the Simulator

In our simulation, each traffic flow is dealt with individually at the link. The simulator uses the next event time advance mechanism to update the simulation clock. Events in the simulation include arrivals and terminations of flows from the three traffic classes, where it is possible with minimal modifications to scale the simulator to analyse the case of more than three traffic classes. The inter-arrival times for arrival events for all three traffic classes follow the Poisson distribution.

After the arrival, the flow is placed on the link for its duration. The duration of the flow depends on whether the size of the flow is determined by the size in bits, which is the case with rate adaptive (stream and elastic) applications, or it is determined by the duration of the connection, which is the case with brittle applications. In both cases, the size/duration of the traffic flow follows the exponential distribution. The reason for choosing the exponential distribution instead of the Pareto for elastic traffic flows will be explained in section 5.4.3.2.

Figure 5.8 shows the mean number of active brittle flows on the link in the fixed partitioning scheme. The initial phase of the simulator is clearly visible. The simulation model is considered to be in the *steady state* after 80,000 simulation seconds. Figure 5.8 shows that this is a valid assumption. The measurements were sampled from 80,000 seconds to 150,000 seconds, following the initial-data deletion technique [LAW91][BOD01].

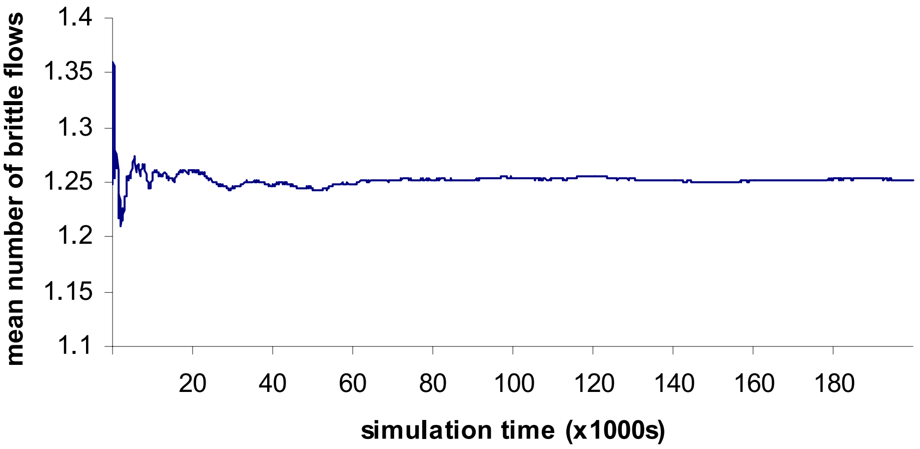

Figure 5.8. Estimation of the Initial Simulation Period

5.4.3. Validation of the Simulation

Before the results of output analysis can be trusted, either as correct for the model or as good measures for the system under study, the simulation model needs to be validated. Validation determines whether the simulation model is an accurate representation of the system under study.

In order to prove the simulation we have built is valid, it is necessary to look further into the analysis of the behaviour of individual traffic classes within the model. When it comes to the resource allocation model, stream and elastic traffic are very similar. The way the bandwidth is allocated to stream and elastic traffic is equal. The main difference lies in the admission control and in the fact that stream traffic flows define the maximal bandwidth level, where for the elastic traffic the maximal bandwidth level is equal to the overall link bandwidth. Therefore, this section analyses two allocation models, one for the brittle traffic class, and the other for the elastic traffic class.

Because of the partitioning, the traffic classes have been observed and analysed independently. The dynamics of the Dynamic Bandwidth Partitioning scheme is omitted in this analysis, since the bandwidth allocation model is of interest, not the final scheme performance. We call this bandwidth allocation scheme Fixed bandwidth partitioning. The features of this scheme will be analysed in more detail in the next Chapter.

The concluding analytical results from this section are compared with the simulation, and are used as the validation of the simulation tool.

5.4.3.1 Brittle Traffic

This section presents the validation for the case of brittle traffic. Brittle traffic flows are determined by their duration time. In the simulation, these traffic flows on arrival are assigned a duration time, which is exponentially distributed around some mean value.

Each brittle traffic flow is allocated a fixed bandwidth b_b. While in the system, these flows occupy this constant amount of bandwidth. For fixed partitioning, the brittle traffic has a fixed portion $\alpha_b B$ of the link bandwidth B available. If all of the available bandwidth is occupied, the incoming brittle traffic flow is rejected at the admission control.

The resulting model for brittle traffic is the M/M/K/K queuing system [GRO85][NUN99], where K is the maximum number of traffic flows we can establish on the link. In the model, traffic flows come as a Poisson stream with intensity λ_b. Flows have exponentially distributed holding time, with mean $1/\mu_b$. The brittle load is

$$\rho_b = \frac{\lambda_b}{\mu_b K}. \qquad (5.13)$$

The M/M/K/K system is a special case of a M/M/C/K system. The maximum number of real-time flows on the link is given by $K = \frac{\alpha_b B}{b_b}$. The probability that there will be *n* flows on the link is

$$p_n = \left[\sum_{i=0}^{K} \frac{(\lambda_b/\mu_b)^i}{i!} \right]^{-1} \frac{(\lambda_b/\mu_b)^n}{n!} \qquad (5.14)$$

The blocking (rejection) probability is easy to calculate:

$$p_K = \left[\sum_{i=0}^{K} \frac{(\lambda_b/\mu_b)^i}{i\,!}\right]^{-1} \frac{(\lambda_b/\mu_b)^K}{K!} \qquad (5.15)$$

In the analysis the average number of brittle flows on the link is calculated. The mean number of active brittle flows is calculated as

$$L = \sum_{n=0}^{K} np_n = \left[\sum_{i=0}^{K} \frac{(\lambda_b/\mu_b)^i}{i\,!}\right]^{-1} \sum_{n=0}^{K} n \frac{(\lambda_b/\mu_b)^n}{n!} \qquad (5.16)$$

In order to validate the simulator for the case of the brittle traffic, we compare the calculated mean number of brittle traffic flows from (5.16) with the mean number of brittle traffic flows in the simulation model. The comparison is presented on Fig 5.9. This result is considered as a validation of our simulator.

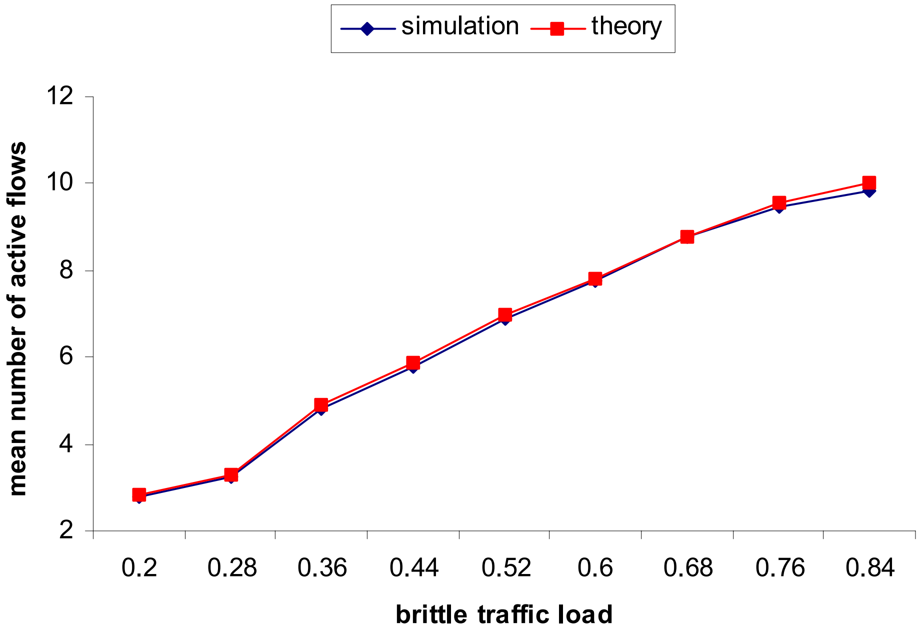

Figure 5.9. Comparison of the Mean Number of Active Stream Flows

5.4.3.2. Elastic Traffic

This section presents the validation for elastic traffic. The main feature of elastic flows is that they are determined by the size in bits, and not by the duration. Their duration in the system depends heavily on the allocated bandwidth.

We consider a model where elastic traffic has a fixed portion of link bandwidth available, $B_e = \alpha_e B$. All active elastic flows get equal portion of the capacity. If there are n_e active elastic traffic flows, they all receive bandwidth $b_e = \frac{B}{n_e}$. Elastic traffic flows arrive as a Poisson stream, with intensity λ_e. Flows have exponentially distributed sizes, with mean f_e. The elastic load on the link is

$$\rho_e = \lambda_e f_e / \alpha_e B \qquad (5.17)$$

The resulting model for elastic traffic is an M/G/1/K queue with so-called *generalised processor sharing*. Nunez-Queija et al [NUN99] analysed the performance of partitioning policies, and defined the theoretical model for elastic traffic flows in two ways. Firstly, if maximum and minimum capacity for elastic flows is defined as $b_{e\min} \neq 0$ and $b_{e\max} < B_e$, i.e. when we have the admission control procedure for elastic traffic, the model results in a M/G/1/K system, where K equals the maximum number of elastic flows in progress.

For this model, the explicit performance results are available in [COH]. For $n = 1, 2, ..., \mathrm{K}$, let $\phi_n = \prod_{j=1}^{n} \frac{1}{b_e} = \left(\frac{n}{B_e}\right)^n$, and $\phi_0 = 1$. Then we can calculate the blocking probability, p_e, and the expected file transfer delay for the file of size *x*, $E[T_e(x)]$ as

$$p_e = \frac{(\lambda_e f_e)^{K_e} \phi_{K_e} / K_e!}{\sum_{j=0}^{K_e} (\lambda_e f_e)^j \phi_j / j!} \qquad (5.18)$$

$$E[T_e(x)] = (x/B) \frac{\sum_{n=0}^{K_e - 1} (\lambda_e f_e)^n \phi_{n+1} / n!}{\sum_{j=0}^{K_e} (\lambda_e f_e)^j \phi_j / j!} \qquad (5.19)$$

Formula (5.19) shows that the mean file transfer delay $E[T_e(x)]$ is proportional to the file size *x*.

If we have a “real” best-effort policy (i.e. $b_{e\min} = 0$ and $b_{e\max} = B$, which is our case), then the model becomes the ‘standard’ M/G/1 processor sharing queue. The processor sharing (PS) queuing discipline was initially introduced by Kleirnock [KLE76]. It has later been thoroughly analysed in the literature [YAS83][COH]. Kleirnock defined it as the limiting case of round-robin scheduling for a time-sharing system in which the time quantum is allowed to approach zero. This discipline assumes [YAS83] that when there are *n* requests in the single-server system of capacity C , each request receives service with rate C/n . The most important feature of the model is that the holding times for elastic traffic connections are not pre-defined, they depend on the state of the link, i.e. on the total number of connections on the link.

This is exactly what we have in our model, n_e elastic connections using available bandwidth for elastic traffic, which is equally shared among active connections. A new connection begins to receive service (a share of the server) immediately after entering the system. Jumps of the service rates take place at the instants when a number of requests in the system change.

By the assumption of equal sharing, the number of flows in progress is geometrically distributed [YAS83, corollary 3.1]:

$$P_r[n \text{ flows}] = \rho_e{}^n (1 - \rho_e) \text{if } \rho_e < 1 \qquad (5.20)$$

It is interesting to note that this result is insensitive to the distribution of document sizes. This is the main reason why an exponential distribution of file sizes has been used instead of a Pareto distribution. A Pareto distribution is usually used when modelling the elastic Internet traffic. It exhibits *heavy-tailed* features which are characteristic for the distribution of the file sizes on the Internet [ARV99][PAR96].

The M/G/1 processor sharing model gives a straightforward calculation of the mean number of active flows on the link,

$$E[n_e] = \frac{\rho_e}{1 - \rho_e} \qquad (5.21)$$

The mean file transfer delay (the average time needed to transfer a file):

$$E[T_e] = \frac{1}{\lambda_e} \frac{\rho_e}{1 - \rho_e} \qquad (5.22)$$

The simulation result for complete partitioning strategy is shown in figure 5.10. The comparison of that result with the values calculated from (5.22) is considered as the validation of the simulation.

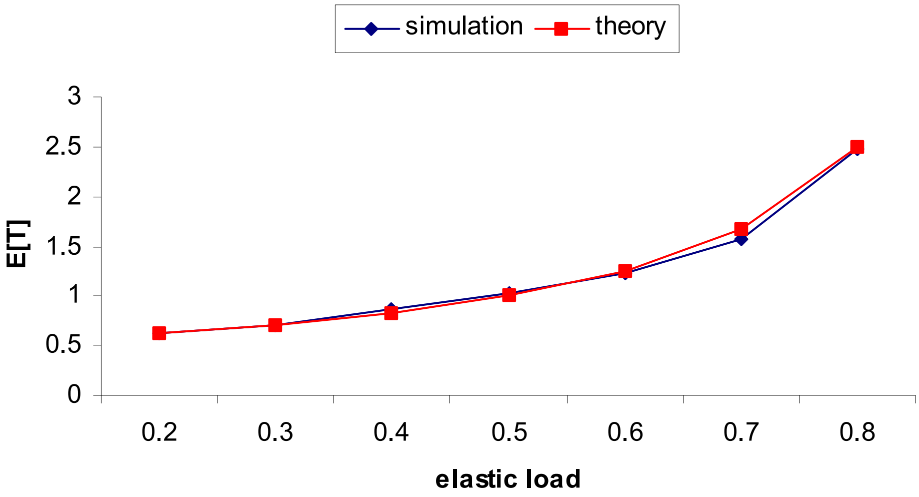

Figure 5.10. Comparison of the Mean File Transfer Delay for Elastic Traffic Flows

5.5. Summary

A purposely-built discrete-event simulator was developed for this research in the C programming language. The simulation is connection-level, designed to analyse different bandwidth allocation schemes in the multi-class networks. The network topology that was analysed consists of a single network link. This Chapter presented the details of the simulation tool built in order to evaluate the efficiency of the Dynamic Bandwidth Partitioning scheme.

In the simulation, each traffic flow is dealt with individually at the link. Events in the simulation include arrivals and terminations of flows from the three traffic classes. The simulator needed to be validated before its results could be taken as a valid representation of the real network events. The validation was done for brittle and for elastic traffic by comparing the output of the simulation with the appropriate theoretical models.

The performance of different bandwidth allocation schemes is modelled using utility functions. Each of the three traffic classes (brittle, stream and elastic) has a utility function defined. This Chapter analysed the three utility functions in detail, defining the way the function shape is modified when the performance requirements change. The brittle utility function cannot be modified, since it is a simple stepwise function, but the stream and elastic utility functions have *shaping parameters* that modify the shape of the function. The impact of the shape of the utility functions on the overall network performance will be analysed in more detail in the next Chapter.

CHAPTER *6*

SIMULATION RESULTS AND ANALYSIS

6.1. Introduction

After defining the Dynamic Bandwidth Partitioning (DBP) scheme and the simulation model in detail in the previous Chapters, this Chapter presents the study of the experimental results. A number of experiments have been conducted in order to study the scheme efficiency and to find the optimal environment for the implementation of the scheme.

There are two general methods for the performance evaluation of a network [PIT00]: measurement techniques and predictive techniques. We are using predictive techniques here, using the network simulator that was described in the previous Chapter. Although the predictive method for performance analysis assumes the mathematical analysis and the simulation, we limit the mathematical analysis to the analysis done to provide the validation of the simulator, and in this Chapter concentrate on the simulation study.

The results presented here discuss the performance and efficiency of the partitioning algorithm, by analysing the impact of different parameters and traffic loads on its performance. Furthermore, this Chapter presents the comparison of the DBP scheme with other bandwidth allocation concepts, namely with the concepts of *best-effort* (full traffic aggregation) and *fixed bandwidth partitioning*.

The simulator gave full freedom in the choice of numerous input parameters, including the traffic load. The way the simulator is designed gives the freedom to select the incoming rate for the traffic flows from the three traffic classes. The simulation data, which are presented as the input to the simulator, are described in section 6.2.

Traffic load defines the amount of traffic that is presented to the network. The calculation of the traffic load for different traffic classes was given in Chapter 5. In the case of stream and elastic traffic, the traffic load does not depend only on their incoming rate, but it is influenced by the network state and traffic loads of other traffic classes. In order to calculate the load stream and elastic traffic put on the network, it is necessary to know the amount of bandwidth available for the traffic classes, which is given in more detail in section 5.4.3. In the DBP scheme these amounts of bandwidth are not constant, but they vary according to the partitioning algorithm. Therefore, we are able to provide only *evaluation* of the load for stream and elastic traffic for Dynamic Bandwidth Partitioning. This is why a brief analysis of the traffic load is given before each of the experiments in this Chapter.

Different performance measurements were used for the performance evaluation of Dynamic Bandwidth Partitioning and other bandwidth allocation schemes. Section 6.3 introduces a new performance metric – *connection utility*. The use of connection utility is a novel way of performance measurement and therefore presents an important contribution of the research presented here. Further to this metric, conventional metrics such as mean link utilisation and mean traffic flow duration were used.

The experiments are grouped into two sections. Firstly, section 6.4 analyses the scheme itself, explaining the reasons for choosing the given parameters and for defining the algorithm in the way it was defined in Chapter 4. Secondly, section 6.5 presents the performance comparison of the scheme with other bandwidth allocation concepts. Each of these two sections are concluded with a summary, so the overall summary of this Chapter is omitted.

6.2. Simulation Data

Simulation was used to evaluate the performance of the scheme. The alternative method would include the measurement of the performance of the real network. That would give us the most accurate information about the network. However, this type of measurement is rarely available, and furthermore there are constraints [PIT00] in measuring the performance of a new scheme, mainly concerning the size of the experimental network.

The simulator and the simulation model have been explained in more detail in the previous Chapter. In this section the data that the simulator used as the input data are presented and explained. When building the simulator, we have had in mind the number of options the simulator needs to provide in order to be able to simulate a large spectrum of network states. Table 6.1 describes the input data to the simulation. We can see that there are three main choices: the incoming rate for the traffic flows, the bandwidth requirements and the prioritisation (scaling parameters). All of these three choices are applicable to each of the three traffic classes.

The first input in the simulator is the simulation time, which was chosen to be 150 000 simulation seconds, so that the time available for sampling the data from the simulation could be large enough, after the deletion of the initial 80 000 simulation seconds [BOD01]. Time interval for partitioning update T_{upd} defines the discrete time intervals in which the partitioning algorithm is performed. The choice for the value for T_{upd} is an interesting problem, which is discussed in more detail in section 6.4.4.

The generation of traffic flows follows a Poisson distribution with a mean equal to the specified number of connections per second. It is possible to define the mean number of connections per second N_i for all three traffic classes, $i \in \{B, S, E\}$. The mean interarrival time T_i for the arrival events in the simulation is then easy to calculate:

$$T_i = \frac{1}{N_i} \qquad (6.1)$$

The connection duration (holding time) depends on the traffic source and on the state of the network link. Simulation data include the mean duration of brittle traffic flows and the mean size of stream and elastic flows. Both of these parameters follow the exponential distribution with the mean values as given in Table 6.1. The reasons why the exponential distribution is used instead of the Pareto distribution have been discussed in Chapter 5.

Table 6.1. Simulation Data

Simulation Time T_{sim} (sec)	150 000
time interval for partitioning update T_{upd} (sec)	240
mean duration for brittle flows t_B (sec)	30
mean size for stream flows f_S (Mbit)	2.0
mean size for elastic flows f_E (Mbit)	4.0
requested bandwidth for brittle flows b_B (Mbit/s)	5.0
requested bandwidth for stream flows min/max $b_{S\min} / b_{S\max}$ (Mbit/s)	0.1/2.0
requested minimum bandwidth for elastic flows $b_{E\min}$ (Mbit/s)	0.01
link bandwidth B (Mbit/s)	100
scaling factor for brittle traffic ω_B	10
scaling factor for stream traffic ω_S	3
scaling factor for elastic traffic ω_E	0.5
decrease parameter ε	0.05

The next simulation parameters are the bandwidth requirements. It has been explained in detail in Chapter 5 how bandwidth requirements influence the shape of the utility functions.

The results in section 6.4.2 will show the effect of the shape of utility function on the overall network performance. Furthermore, after the introduction of the admission control, the choice of the minimal bandwidth requirement has a large impact on the network load.

There is no particular reason why some other similar values have not been chosen for the simulation data. Throughout this Chapter, *traffic load* will be used as the evaluation of the amount of traffic on the link. The information about the traffic load includes both the bandwidth requirements and the mean sizes and durations.

The choice of the link bandwidth B is arbitrary and is not connected to any of the existing transmission systems. Experiments with various link bandwidths are presented in section 6.5.4. It is interesting to assess the performance of different bandwidth allocation concepts in the presence of various available link capacities.

The scaling parameters ω_B, ω_S and ω_E are very important. They define relative priorities given to traffic classes. We used a number of different combinations for scaling parameters in our experiments, which will be explained in more detail in section 6.5.5.

Finally, the decrease parameter ε defines the dynamics of the scheme. The choice of this parameter is extremely important. Section 6.4.3 analyses the impact of different values for the decrease parameter.

6.3. Performance Evaluation Metrics

Before we start to explain the experiments and the simulation results, it is important to discuss the performance measures (metrics) that have been used. In general the measures of interest depend on the system and are invariably used to indicate its predicted performance under certain conditions. However, when it comes to telecommunication systems, there are several basic performance measurements. These include:

- delay
- delay variation
- loss
- throughput

However, the research presented here observed the problem of optimal allocation of the network bandwidth between the active *traffic connections.* This is why the connection-level simulator has been developed, and the performance measurement that we are interested in is on the connection level. This does not include the measurement of the delay variation and loss probability, since these are the features of packet-level analysis.

Delay was measured in our experiments in the form of the *mean flow duration* for elastic traffic. The mean flow duration is the average time needed for the elastic traffic flow to be transported through the network. The issue of end-to-end delay on a multi-link network model, however, is a very important performance measurement that needs to be included in the future analysis of the Dynamic Bandwidth Partitioning concept. Chapter 7 discusses future research issues and analyses the ways of introducing the end-to-end delay in the utility evaluation. For the experiments presented in this Thesis, since the observed network model consists of a single link, the end-to-end delay has not been included in the analysis.

When it comes to bandwidth allocation, various authors have proposed other performance metrics. Breslau and Shenker [SHE98] use *bandwidth gap* to assess the efficiency of reservation-capable and best-effort networks. The bandwidth gap shows the amount of bandwidth that needs to be added or subtracted from the network link in order to produce the same performance in terms of the traffic utility for different bandwidth allocation concepts.

Steenkiste and Ma [STE00] in their QoS-based routing algorithm, calculate the link cost by using *virtual residual bandwidth* that captures the congestion conditions on the network.

Since Dynamic Bandwidth Partitioning is based on the notion of utility, the most natural conclusion is to evaluate the performance of the scheme by using *utility* as the evaluation metric. We are observing a new evaluation metric that is named ***connection utility***.

Connection utility is calculated when the traffic flow terminates, by simply calculating the time average utility while the flow was active. For analytical purposes, we can approximate the calculation of this average with an integral in equation (6.2). The connection utility v_{ij} of a traffic flow i that belongs to the traffic class j can be seen as the approximation of the network performance that the traffic flow received while in the network.

$$v_{ij} = \frac{1}{T_{dur}} \int_{0}^{T_{dur}} u_j\left[b_i(t)\right] dt \qquad (6.2)$$

T_{dur} is the time traffic flow spent on the link.

We observe the mean connection utility for a large number of flows and use it as a comparison parameter for comparing the Dynamic Bandwidth Partitioning scheme with other bandwidth allocation schemes. The connection utility was defined with the objective to give us the most accurate evaluation of the network performance in the multi-class system, where traffic classes require different treatment. Section 6.5 brings the performance comparison between different bandwidth allocation concepts on the basis of mean connection utility, mean flow duration and mean link utilisation. It is very interesting to see how observing different performance metrics can lead to very diverse conclusions. Table 6.2 provides the summarised comparison of the performance metrics used for the experiments.

It is important to note that, for bandwidth allocation schemes that deploy admission control, each flow rejection generates a negative utility $v_{ij} = -\omega_j u_j \lfloor b_j(t) \rfloor$, where $b_j(t)$ is the bandwidth allocated to the traffic flows belonging to traffic class j at the moment of the rejection of the incoming flow. When a brittle traffic flow is rejected at the admission control point, this rejection generates a negative utility of $-\omega_B$.

Table 6.2. Comparison of Performance Metrics

Metric	***Explanation***	***Why is it a good metric***	***Why is it a bad metric***
Average Connection Utility	Average Level of end-users' satisfaction with the network performance	Takes into account the quality of service and the traffic heterogeneity	Not accurate enough, gives just a general overview of the users
Link Utilisation	The average amount of link bandwidth in use	Shows the average usage of the network resources	Does not consider traffic differentiation, bandwidth requirements or QoS
Average File Transfer Time	The average time for a file to be transported in the network	Shows very precisely the congestion level in the network	Cannot be used for the brittle traffic

6.4. Performance Evaluation of the Dynamic Bandwidth Partitioning Algorithm

6.4.1. Introduction

The Dynamic Bandwidth Partitioning scheme has been defined in detail in Chapter 4. We have seen that the main aspects of the scheme are the partitioning algorithm and the bandwidth allocation. In this section we analyse the reasons for defining the scheme in the way it was defined in Chapter 4.

Before we compare the performance of the scheme with other bandwidth allocation schemes, it is necessary to analyse the details of the partitioning algorithm – to find out the optimal decrease parameter ε; to analyse the impact of different utility functions and bandwidth requirements on the dynamics of the algorithm; to compare the scheme performance for various time updates. The following experiments study these and similar issues.

6.4.2. The Impact of Utility Functions

In this section we analyse different parameters for utility functions for stream and elastic traffic. By changing the shaping parameters a_{s1}, a_{s2}, and a_e in the utility functions (5.2) and (5.9) we want to see the impact of different shapes of functions (5.2) and (5.9) to the overall network performance. The objective of this experiment is to see how different utility functions influence the partitioning algorithm.

It is important to note at this point that the shaping parameters depend on the performance requirements of the applications (or end-users). Therefore, these experiments will show whether Dynamic Bandwidth Partitioning scheme is adaptive to different performance requirements.

We cannot expect that the network operators will use a single utility function to model the performance requirements of end-user applications. The network operators are given a choice of shaping parameters which they can use for modelling. Therefore, the change of the shaping parameters should not be arbitrary. On the basis of the agreements between the network operators and the end-users, or on the basis of different bandwidth requirements by the end-users, different utility functions will be used.

This is also true for the brittle traffic. It was already discussed in Chapter 5 that Sarkar and Tassuilas [SAR99] used a stepwise function with a number of non-zero levels for the utility function of video traffic. The analysis of such a function is out of the scope of this Thesis, since it includes the analysis of the nature of the video traffic. Here we focus only on the possibility for different shapes of utility functions for stream and elastic traffic.

6.4.2.1. Impact of the Changes in the Elastic Utility Function

The utility function for the elastic traffic is concave, with a very simple mathematical expression:

$$u_e(b_e) = 1 - e^{-\frac{a_e b_e}{b_{e\max}}} \qquad (6.3)$$

It is interesting to see what happens with the overall performance of the scheme when parameter a_e changes. Figure 5.7 shows the impact of the shaping parameter a_e on the function shape.

By decreasing a_e the network operator can model the case when the elastic traffic flows are less interested in large bandwidth. For example, this can be a case of a delay-insensitive application, such as transfer of files where delay is of less importance. If we consider a future network where the services will be charged according to the performance, we can assume that the users will be ready to save some money even if that means receiving worse network performance. There is no need to model the behaviour of such users by a steep utility function.

The objective of the experiment in this section is to show that the users with smaller requirements will really receive worse performance. Showing that this is true, we would prove the adaptability of the Dynamic Bandwidth Partitioning scheme.

A number of experiments have been done with different values for a_e while all other parameters, including the scaling parameters and two other utility functions have been kept constant. The measurements have been done for different traffic loads, in order to find out the impact of the shaping parameter in various congestion environments. The examples of the traffic load scenarios are given in table 6.3. The loads used for this experiment show that we have used a large load for the elastic traffic, in order to find the case when the impact of the elastic utility function is most likely to be seen. Other simulation data is given in table 6.1.

Table 6.3. Load Patterns for the Experiment with Elastic Utility Functions

Load Scenario	Brittle Load	Stream Load	Elastic Load
1	0.04	0.413	0.165
2	0.08	0.51	0.275
3	0.10	0.54	0.360
4	0.14	0.689	0.650

In order to assess the dynamics introduced by using different utility functions, the average file transfer time for the elastic traffic is measured. Since average utilities are calculated by using different utility functions, the average file transfer duration is the best choice of the performance metric in this case.

The second parameter that has been measured in this experiment is the number of partitioning updates. The partitioning update happens when the change in the weighted normalised utility changes for a substantial amount, as presented in Chapter 4. This measurement will show the dynamics of the partitioning algorithm, and how different utility functions influence this dynamics.

The result is given on figure 6.1. We know from section 5.3.3 that the parameter a_e is defined by the shaping rule, θ_e, according to (5.12). The analysed shaping rules are presented in table 6.4. We are observing three different shaping rules and therefore three different values for the parameter a_e.

Table 6.4. Shaping Parameters a_e

Shaping Rule θ_e	a_e
0.75	1.3863
0.9	4.6052
0.95	5.9903

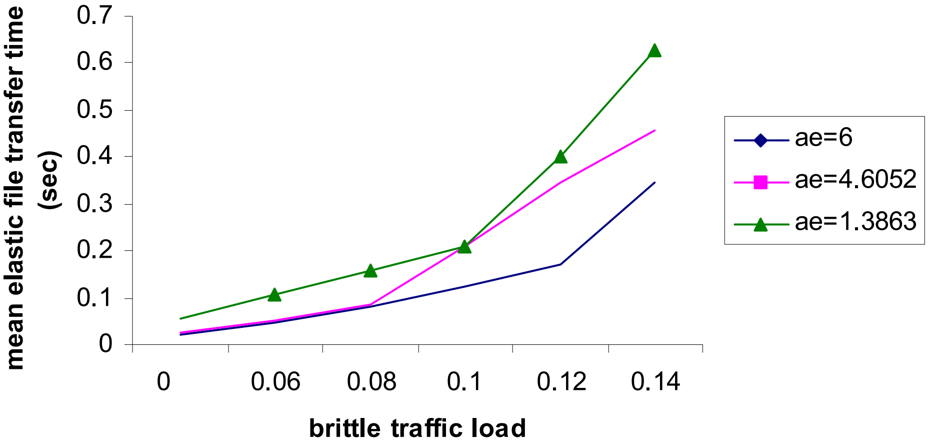

Figure 6.1. Impact of the Shaping Parameter a_e on the Network Performance

The result in figure 6.1 is very interesting, since it shows that the average file transfer time increases with the decrease in the parameter a_e. The increased parameter a_e means a less rigid utility function, as we can see on figure 5.7 in the previous Chapter. This means that the more rigid requirement elastic traffic flows put on the network, the performance they get will be better.

However, this conclusion depends heavily on the network environment. In the experiment above, scaling and other two utility functions were kept constant. Only the elastic utility function has changed. If other circumstances changed on the network, the influence of different elastic utility functions would be less obvious.

Nevertheless, this result is very interesting and very important for the performance analysis of the DBP scheme. It shows that the choice of utility function is very important.

When it comes to how the dynamics of the partitioning algorithm is affected by different elastic utility functions, the results are given in table 6.5. The load scenarios 1-4 are given in table 6.3. We can see from table 6.5 that the bigger the parameter a_e is, the fewer the updates of the partitioning parameter. This is an expected result, since smaller a_e means less steep function which means that the range of values for the utility of elastic traffic is larger, giving more space for the partitioning updates. Also we can see that as we increase the load the number of updates decreases sharply. This is due to the fact that in the congested network Dynamic Bandwidth Partitioning works very much as the fixed bandwidth partitioning, since the load is very high, and it is unlikely that the changes in the normalised utility will be large enough.

Furthermore, the case with heavy load is that usually the brittle traffic takes all available space, leaving the stream and elastic traffic to use only the minimal space they are allowed to use. That environment is unlikely to change in the congested network. It would take a large burst of stream traffic flows and consequently a large decrease in the stream utility to change anything in a congested network. That is why the number of partitioning updates decreases with the increase in the traffic load.

Table 6.5. Impact of the Shaping Parameter a_e on the Partitioning Dynamics

Parameter	Number of Partitioning Updates for Various Load Scenarios			
	Load 1	Load 2	Load 3	Load 4
$a_e = 1.3863$	139	47	10	8
$a_e = 4.6052$	55	14	8	7
$a_e = 5.9903$	47	41	23	7

6.4.2.2. Impact of the Changes in the Stream Utility Function

The stream utility function (5.2) has two shaping parameters, a_{s1} and a_{s2}. Their values are not arbitrarily chosen, but they are defined by the maximal and minimal bandwidth requirements $b_{s\min}$ and $b_{s\max}$, and by the shaping rule θ_s, as explained in section 5.3.2.

The experiments that have been done for this section are similar to the ones for the elastic utility function. The measurements of interest is the average bandwidth allocated to the stream traffic flows.

We consider three cases, A, B and C. The cases are described in Table 6.6. For each of the cases the experiments have been conducted for a number of traffic load scenarios. Three examples of the traffic load scenarios are given in table 6.7. The traffic loads were chosen so that the load of stream traffic is much greater then the load of elastic traffic, which in turn is larger then the brittle load. By choosing this load structure, we wanted to see what happens when it is most likely that the difference in the stream utility function will make an impact. Other simulation data is given in table 6.1.

Table 6.6. The Cases for the Experiments with Stream Utility Functions

Case	Requirements	Parameters
A	$b_{s\min} = 0.1$ $b_{s\max} = 2$ $\theta_s = 0.5$	$a_{s1} = 0.8213$ $a_{s2} = 0.1849$
B	$b_{s\min} = 0.1$ $b_{s\max} = 2$ $\theta_s = 0.9$	$a_{s1} = 2.4436$ $a_{s2} = 0.0612$
C	$b_{s\min} = 0.4$ $b_{s\max} = 2$ $\theta_s = 0.9$	$a_{s1} = 8.8567$ $a_{s2} = 2.8464$

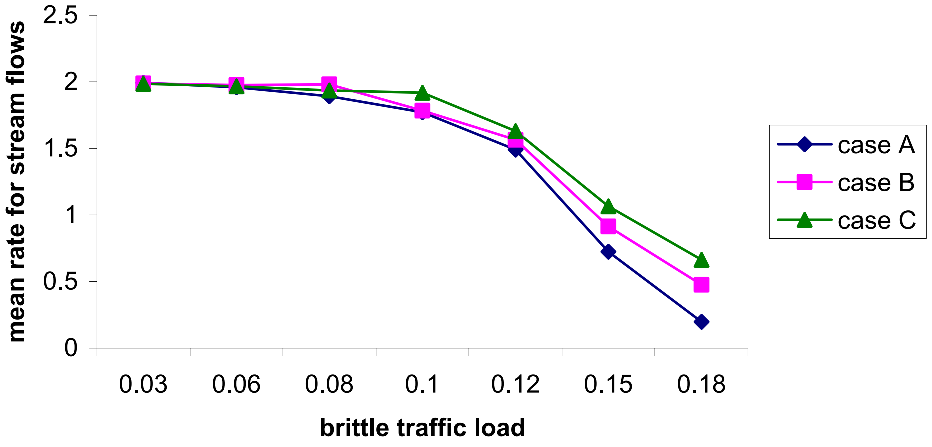

Figure 6.2. Impact of the Shaping Parameters a_{s1} and a_{s2} on the network performance

Table 6.7. Load Patterns for the Experiment with Stream Utility Function

Brittle Load	**Stream Load**	**Elastic Load**
0.04	0.469	0.142
0.08	0.723	0.417
0.12	0.895	0.500

We can see on figure 6.2 how the mean allocated rate for the stream flows is the largest for case C, which is the case where the requirements are most rigid. This shows that the Dynamic Bandwidth Partitioning scheme is adaptable to the traffic profiles and performance requirements. Case A, where the requirements are the least rigid, where it is not that important that the bandwidth is allocated, and the function is not that steep, results in poorer performance.

If we try to assess the impact of different utility functions, it is evident that for higher traffic loads the margin is rather large. Using bandwidth requirements and the utility function from case A results in an average bandwidth level of about 0.2 Mbps, which is a small value compared to the average bandwidth of 0.75 Mbps for the case C. Therefore, the impact of using different utility functions is substantial.

For small traffic loads the impact of the utility function is negligible, since for smaller traffic loads there is no need to perform the partitioning updates.

6.4.3. The Impact of the Decrease Parameter ε

The definition of the Partitioning algorithm in Chapter 5 showed that the increase and decrease in partitioning parameters are defined by a single parameter, which is denoted ε.

This parameter is very important. It defines the speed of the partitioning updates. Therefore, it has a large impact on the speed of the bandwidth allocation updates, which can prove to be crucial for the network performance, especially in the cases of heavy congestion. In this section we will give results of experiments done in order to establish some conclusions about the optimal values of the parameter ε.

Whatever the results of these experiments are, and whatever is the conclusion, it is important to understand a trade-off between the introduced improvement and the implementation difficulties. The problem that can arise is that, when there is a large ε, which is then followed by a large increase/decrease in the bandwidth allocation for the traffic classes, this can result in large blocking rates. If the capacity allocated for a particular traffic class suddenly decreases by a large amount, it could lead to the saturation of the buffer used by that traffic class. This and similar incidents need to be taken under consideration before any conclusion about the optimal values for ε is reached.

However, the precise analysis of this trade-off is out of the scope of this Thesis. Although it is very important and interesting for the research presented here, it involves a packet-level analysis of the dropping rate caused by the sudden changes in the service rate for individual buffers. This can prove to be an interesting subject for future research on the implementation issues of Dynamic Bandwidth Partitioning.

The experiments presented in this section have been done for the traffic load scenarios which are given in table 6.8 in the next section. Other simulation data is given in table 6.1. In the first experiment, we have kept the value of the decrease parameter ε fixed and changed the traffic loads in order to see the impact of ε on the performance of the scheme.

The result in figure 6.3 shows that, when it comes to brittle traffic, it is more optimal to change the partitioning parameters for a smaller value. However, that value should not be too small, as we can also see on figure 6.3, since the effect of the dynamic partitioning is then similar to the effect of the fixed partitioning. The conclusion from this experiment is that neither too large (0.1) nor too small (0.001) values for ε are optimal for the brittle traffic.

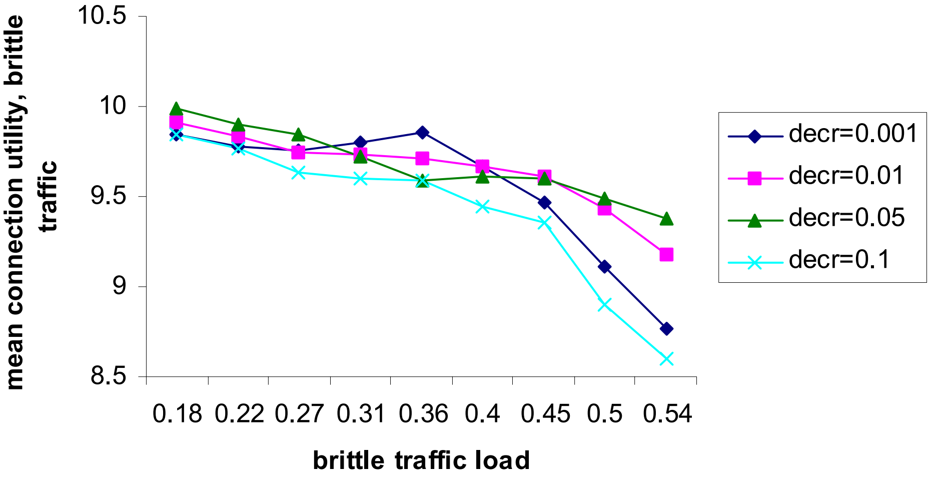

Figure 6.3. Impact of ε on the Connection Utility for Brittle Traffic

When it comes to stream traffic, figure 6.4 shows that the impact of the parameter ε on these traffic classes is completely opposite. The mean connection utility for the stream traffic is greater for very small and very large values of the parameter ε. The reason for this is twofold. Firstly, large ε corresponds to large and sudden changes to the allocated bandwidth. The 'elasticity' of the stream flows and their ability to seize all available bandwidth is important for large ε and that is the reason for better performance with larger ε. Secondly, when it comes to small value for ε ($\varepsilon = 0.001$ here), this case corresponds to *fixed* bandwidth partitioning, which will prove to be an efficient scheme for the stream flows. The reason for this is in the limited interference with other traffic classes. Performance of the fixed bandwidth partitioning concept will be studied in more detail in section 6.5.

Finally, figure 6.5 shows that elastic traffic receives very poor performance for small value of ε, because its elasticity cannot be fully utilised in a scheme which shows little or no dynamics. Elastic traffic needs larger ε, because larger ε provides larger fluctuations in the available bandwidth.

It is not easy to draw a single conclusion when it comes to the optimal level of dynamics of the partitioning algorithm. Once more the heterogeneity of the traffic and the diversity of requirements traffic classes out on the network becomes obvious. While brittle traffic prefers medium values for ε, stream traffic prefers critical values, large or small, and elastic traffic finds large values for ε optimal.

The optimal decrease parameter ε, therefore, needs to be chosen depending on the prioritisation, traffic loads, and the implementation.

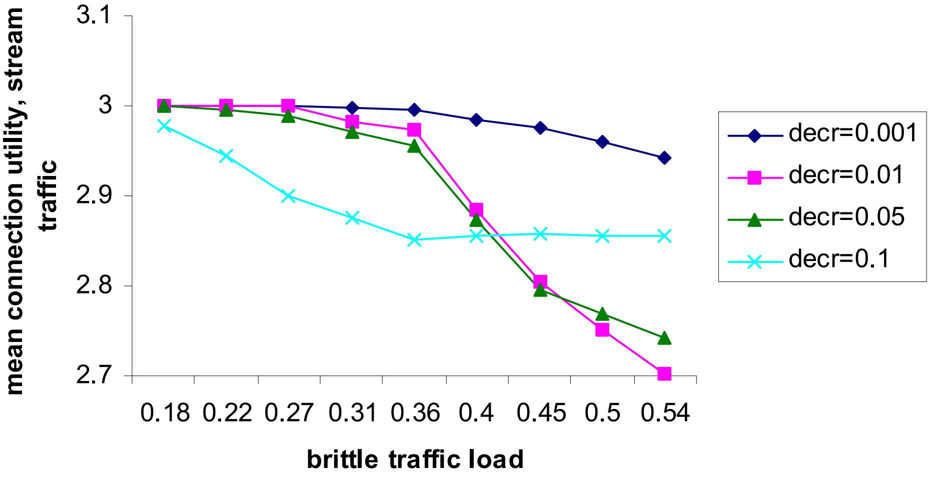

Figure 6.4. Impact of ε on the Connection Utility for Stream Traffic

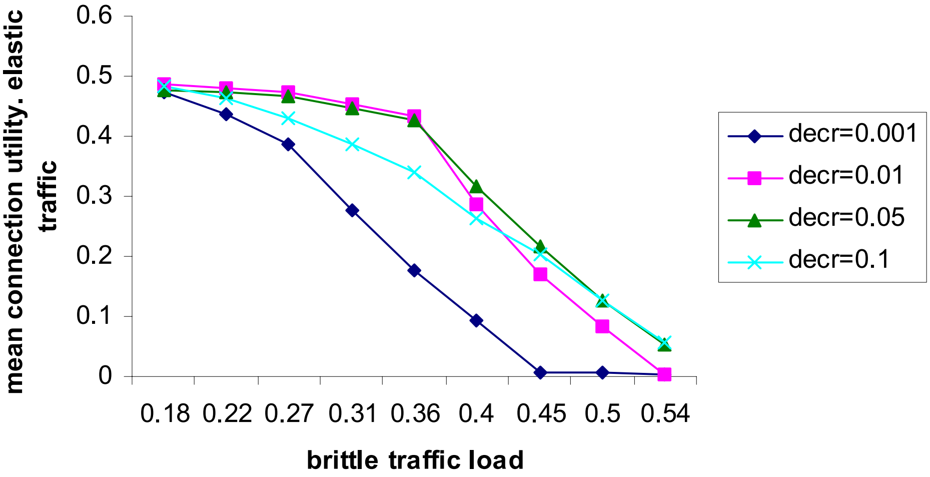

Figure 6.5. Impact of ε on the Connection Utility for Elastic Traffic

Another interesting result is shown on figure 6.6. In this experiment, the traffic load is kept constant (medium traffic load, approximately 0.45 for all three traffic classes), while the decrease parameter ε changed its value from 0.01 to 0.1. This is the 'middle' range of the values for ε. It is interesting to note that the mean connection utility does not change much in the observed interval, it is only when ε becomes close to 0.1 that there is a small decrease in the mean connection utility for all three traffic classes. This shows that it would be almost equal to choose any value for ε from the 'middle' range of values.

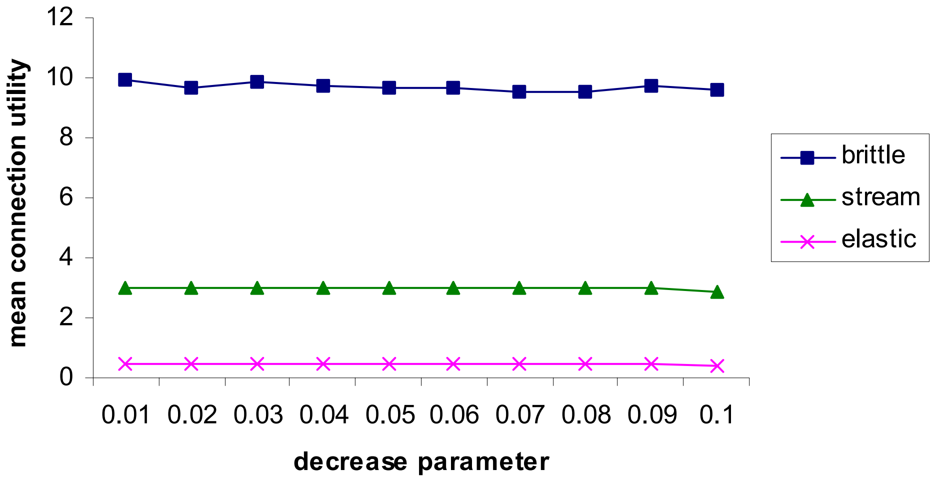

Figure 6.6. Analysis of the Impact of ε

6.4.4. Analysis of Partitioning Update Intervals

The Partitioning algorithm is performed at discrete time intervals. In the first versions of the DBP scheme [RAK5] the algorithm was repeated after each *event* on the network, i.e. after each arrival or departure of a new traffic flow. The problem with this approach is that it is not scaleable, since the partitioning algorithm is performed too often. This implies large overhead and ineffective mechanism, because the bandwidth partitioning parameters are constantly changing.

That is why the current version of DBP uses discrete time intervals to check whether there is a need for a partitioning update.

The most important question when it comes to the partitioning update is, how often should it be performed? What is the optimal time interval for the partitioning algorithm?

The experiments shown in this section give guidance towards the choice of an optimal update interval. The network performance has been checked in three cases: when the interval was small ($T_{upd} = 60$ simulation seconds), medium ($T_{upd} = 240$) and large ($T_{upd} = 1000$). These values for T_{upd} are arbitrary, $T_{upd} = 240$ was the value that was used in majority of the experiments, while the other two values were chosen to illustrate smaller and larger update intervals. The experiments have been done in the load environment as given in table 6.8 in the next section. Other simulation data is given in table 6.1.

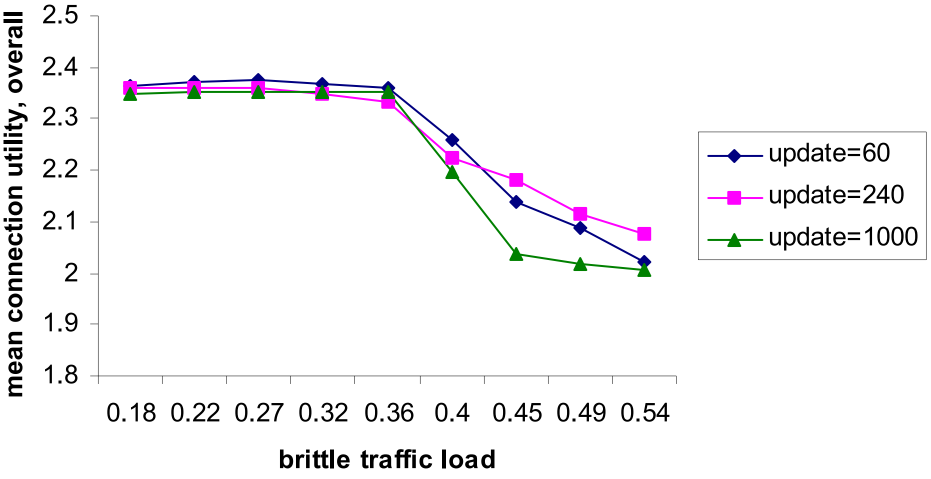

Figure 6.7. Comparison of the Overall Mean Connection Utility for Different Time Updates of the Partitioning Algorithm

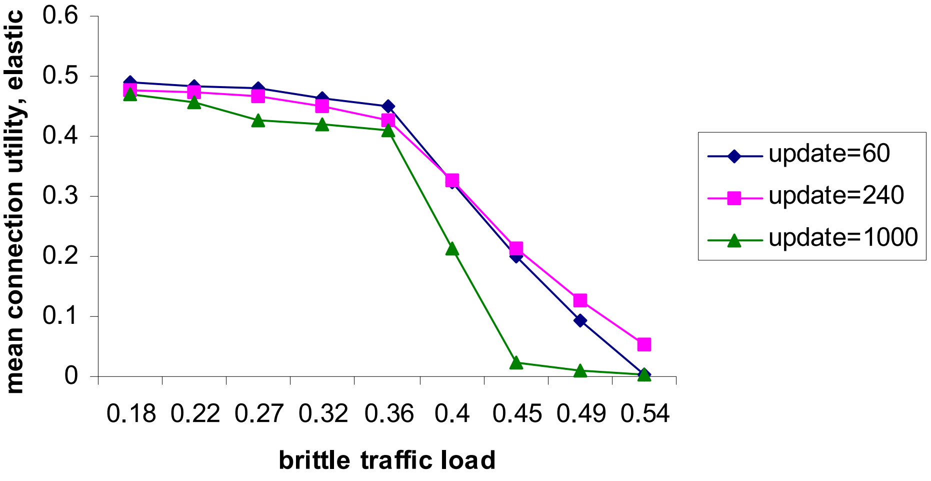

Figure 6.8. Comparison of the Mean Connection Utility for Elastic Traffic for Different Time Updates of the Partitioning Algorithm

We can see from the results on figures 6.7 and 6.8 that the best choice for the update interval is the medium value, since both too small and too large values produce lower mean connection utility. Figure 6.7 shows the overall mean connection utility. We can see that for smaller traffic loads the chosen update intervals perform approximately equally. It is for larger traffic loads that the difference becomes obvious. Figure 6.8 shows the mean connection utility for the elastic traffic only. Large update intervals cannot increase the performance of the elastic traffic, since this type of traffic is very dynamic, and prefers smaller update intervals. Really interesting is the fact that a medium interval of 240 simulation seconds showed a very small performance increase when compared with the small interval. Similar results exist for the stream and brittle traffic, in each of the cases it is the medium update interval that provides the best performance.

6.4.5. Summary

This section provided detailed discussion of experiments examining a number of issues related to the actual design of the Dynamic Bandwidth Partitioning scheme.

Firstly, the impact of the stream and elastic utility functions was studied. The choice of the utility function is important because the utility function is used to model the end-user performance requirements. It was shown in Chapter 5 that for different bandwidth requirements, the derived end-user satisfaction is modelled with different utility functions. Only changes in the utility functions for stream and elastic traffic were studied in this section. The experiments showed that the users who chose less rigid performance requirements really do suffer because of that, since they receive worse performance. This is an interesting conclusion, proving the adaptability of the DBP scheme to variable user requirements. Furthermore, the choice of utility functions is an important control mechanism for the network operators who can define numerous traffic classes and model them using different utility functions.

The next interesting issue that was studied in this section is the decrease parameter ε in the partitioning algorithm. The partitioning algorithm uses additive increase, additive decrease linear control to change the partitioning parameters. The decrease parameter is the step value for the linear control algorithm. It is a very important parameter for the scheme performance. A large decrease parameter causes rapid changes in the bandwidth partitioning and therefore changes the network state and bandwidth allocation substantially. The experimental results show that different traffic classes prefer different values for the decrease parameter. While brittle traffic requires medium values for the decrease parameter, both stream and elastic traffic require critical values, large or small. This is an expected conclusion, since the elasticity of the rate-adaptive applications (both stream and elastic) is better utilised in the network with more dynamic bandwidth allocation. Interesting analysis for the future would be the impact of the decrease parameter on the loss rate due to buffer overflow. However, this would include a packet-level analysis of the scheme performance. Based on the conclusions presented here, the value for the decrease parameter has been chosen to be $\varepsilon = 0.05$ for the majority of experiments.

Finally, the analysis of the optimal time update parameter for the discrete-time partitioning algorithm showed that for all three traffic classes it is the medium value of the time update parameter which is optimal. A large time update parameter is not able to follow the changes in the network state and is therefore not an optimal solution. On the other hand, the small time interval for the update means that the updates are happening very frequently, often more frequently than the changes in the network state. For small time intervals the partitioning updates do not always represent the events on the network. This decreases the average network performance.

The experiments presented in this section conclude the formulation of the Dynamic Bandwidth Partitioning scheme. The next section addresses the performance analysis of the scheme and the comparison with other bandwidth allocation concepts.

6.5. Performance Comparison with other Bandwidth Allocation Concepts

6.5.1. Overview

In order to assess the performance of the DBP scheme, it is necessary to compare it with other bandwidth allocation schemes. The goal of the experiments presented in this section was to simulate different bandwidth allocation concepts in different network environments in order to find the traffic pattern and the network environment in which DBP outperforms other bandwidth allocation concepts. The schemes are compared on the basis of mean link utilisation, mean flow duration for elastic traffic flows, and mean connection utility.

Two other bandwidth allocation concepts were used for the comparison. They are presented in more detail in section 6.5.2.

Schemes are analysed using a number of different traffic load patterns and different prioritisation mechanisms (scaling constants). Sections 6.5.3 and 6.5.4 provide a general comparison between the schemes, through experiments with approximately equal traffic loads. For each individual experiment the loads for all three traffic classes have been increased by increasing the mean rate of traffic flows arriving to the network. Table 6.8 shows the traffic loads in the DBP scheme. The first three columns show the selected values for the mean number of arrivals for each of the traffic classes, and the last three columns show the traffic load. The traffic load can be calculated by using the analysis from section 5.4.3. The input values from table 6.8 have been used for experiments which will be analysed in the next two sections.

Section 6.5.5 studies the performance of the schemes in the environment of traffic fluctuations, when the loads of the three traffic classes are not equal, but when one of the traffic classes puts larger load on the network. It is interesting to observe how different bandwidth allocation schemes react to sudden changes in the traffic load.

Finally, towards the end of this Chapter we present some additional experiments. These include the study of the behaviour of the bandwidth allocation schemes for variable link capacities, the assessment of the negative impact of introducing the admission control, and a very important study of the impact of different scaling schemes. Scaling provides means to network operators to establish the relative priorities among the defined traffic classes.

Table 6.8. Values for Traffic Load

Arrivals per second, brittle	Arrivals per second, stream	Arrivals per second, elastic	Load, brittle	Load, stream	Load, elastic
0.12	3	1.5	0.18	0.337	0.216
0.18	3.25	1.775	0.27	0.406	0.299
0.24	3.50	2.05	0.36	0.418	0.441
0.30	3.75	2.325	0.45	0.609	0.64
0.36	4	2.6	0.54	0.704	0.915

6.5.2. Schemes Used for Comparison

There are two bandwidth allocation schemes that have been used for the performance comparison with the Dynamic Bandwidth Partitioning scheme: Best-effort and Fixed Bandwidth Partitioning.

Best-effort (complete sharing) is the scheme without resource reservation and admission control. All traffic flows are accepted to the network and they all receive equal share of the network capacity. More formally, let us consider a single network link of bandwidth B. If we denote the number of active traffic flows of the traffic class i as n_i, each traffic flow on the link will be allocated bandwidth b:

$$b = \frac{B}{\sum_i n_i} \qquad (6.4)$$

The only limitation in our model is that both brittle and stream traffic flows have the maximum rate defined, and therefore are never allocated more bandwidth then their maximum rate. The remaining bandwidth is then equally shared between the active elastic traffic flows.

It has been discussed earlier in Chapter 3 that this type of bandwidth allocation is optimal for a single-service network, designed for file transfer as the only application. In a number of experiments the comparison of this scheme with Dynamic Bandwidth Partitioning will be used to show the advantages of implementing a dynamic control mechanism. We explore the possible traffic patterns and traffic load environments in which dynamic control will increase the network performance compared to a simple best-effort concept.

Fixed Bandwidth Partitioning allocates the bandwidth in the same way as the Dynamic Bandwidth Partitioning scheme , but the dynamic partitioning algorithm is not being used, i.e. the partitioning parameters are fixed all the time, defined to be equal, $\alpha_B = \alpha_S = \alpha_E = 0.33$. When comparing the performance of this scheme with the DBP scheme, we want to show the benefits of introducing the dynamics into the concept of bandwidth partitioning.

6.5.3. Comparison of Mean Link Utilisation and Mean Flow Duration

Both link utilisation and flow duration have been analysed in section 6.3. Mean link utilisation is defined as the mean amount of bandwidth being used on the network. This metric shows which scheme is able to use more of the available bandwidth space. Mean flow duration is defined as the mean duration of a flow showing rate-adaptive features. The duration for brittle flows in pre-defined and exponentially distributed.

The purpose of the experiments presented here is to analyse and compare different schemes based on the conventional metrics, before we move on to the comparison on the basis of the new metric, mean connection utility.

The comparison of the average link utilisation is shown in Figure 6.9. The caption under the x-axis on this figure is ‘brittle traffic load’. It is important to note that it is not only the brittle traffic that is being increased in this experiment. Stream and elastic traffic, too, are increased following the pattern from table 6.8.

Because the link bandwidth is partitioned, and each traffic class is restricted to using only a limited portion of the link bandwidth, the elasticity of the elastic and stream traffic will not extend to its maximum and the mean link utilisation for the best-effort concept is higher.

It is interesting to note the performance of the fixed partitioning scheme. It has the worst performance, since it is unable to adapt to the network state. The rigid nature of the fixed partitioning scheme decreases the utilisation. While DBP is able to achieve reasonably high link utilisation of 70-75%, fixed partitioning is unable to reach more then 45% utilisation.

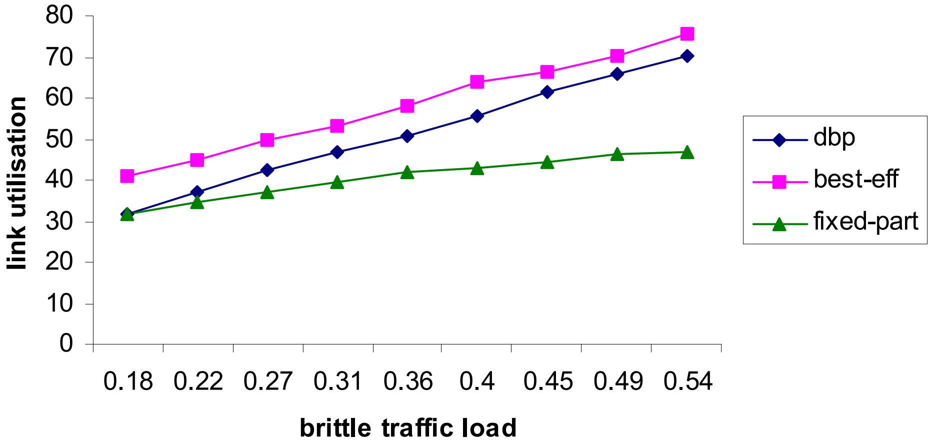

Figure 6.9. Comparison of the Average Link Utilisation

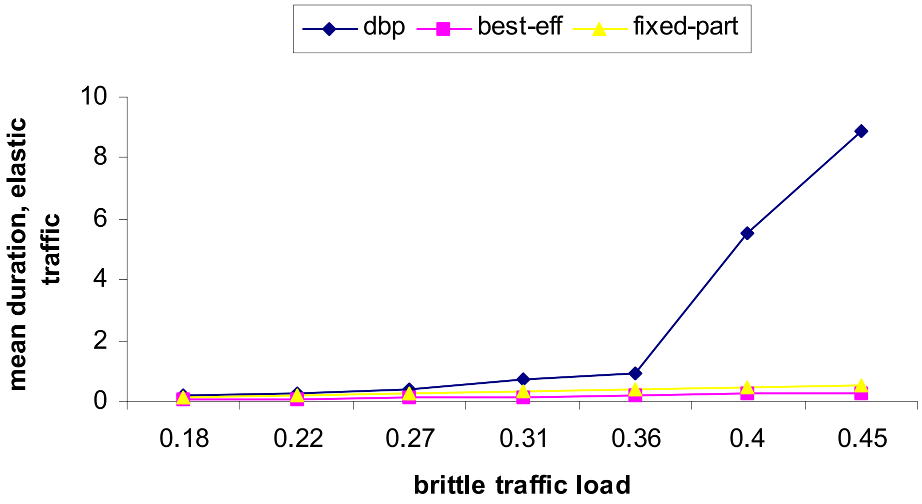

Figure 6.10. Comparison of the Average File Transfer Time for Elastic Traffic

Figure 6.10 provides the comparison of the mean duration (file transfer time) of elastic traffic flows. DBP was created to provide bandwidth guarantees, i.e. to protect the customers that require such guarantees. As we will see later in this Chapter, the DBP scheme is able to do that. Since we are considering constant link capacity, elastic traffic must suffer from the introduction of the new control mechanism, in order for the appropriate level of capacity to be given to the users of brittle applications.

Figure 6.10 shows that the mean flow duration is much longer for the DBP scheme. The difference between DBP and other two schemes is rather large. However, we will be fully able to understand this difference only after analysing the real impact of the larger flow duration. That *real* impact is modelled by utility functions. The flow duration (or indeed the level of bandwidth that is allocated) does not linearly map onto the users' satisfaction – the relation is more complicated.

The results presented here are very interesting. They show the drawbacks of the partitioning concept. However, these results were expected. We knew from before that bandwidth partitioning will not make things better for the elastic traffic, since best-effort Internet was designed especially for elastic traffic; it is an optimal concept for serving the file transfer-based applications.

These results are also interesting to observe together with the comparison on the basis of connection utility, which will be presented in the next section. By observing only conventional performance measurements (mean link utilisation and mean duration), it is possible to draw wrong conclusions. This is only one of the reasons why we decided to analyse the new performance metric, connection utility, and to try to assess the performance with taking under consideration the heterogeneity of the traffic and, above all, the performance requirements and end-user feedback.

6.5.4. Comparison of Mean Connection Utility

Connection utility was explained in more detail in section 6.3. The mean connection utility is calculated by adding all the individual connection utilities during the simulation time, and dividing them by the number of traffic flows that have been active on the simulated network during that time.

We compare both the *overall* and *per-class* mean connection utility in this section. The calculation of the overall mean connection utility includes traffic flows from all traffic classes. The accuracy of this calculation is doubtful. Elastic traffic flows are relatively small in size (mean size 4Mbit). The number of elastic traffic flows, therefore, needs to be big in order to achieve similar traffic load as for the brittle flows that are rather long in duration (mean 30 sec).

This is why it is very important to analyse the mean connection utility for each individual traffic class. The overall connection utility is the only way to incorporate the conclusions from the per-class comparisons, and we are using it here because of that.

The simulation data for the experiments presented here is given in table 6.1, and the traffic loads are as in table 6.8.

The result is shown on figure 6.11. As the traffic load increases, DBP shows slightly better performance than the best-effort scheme. This becomes more obvious for higher traffic loads. Fixed partitioning performs poorly, generating lower connection utility for both small and large traffic loads.

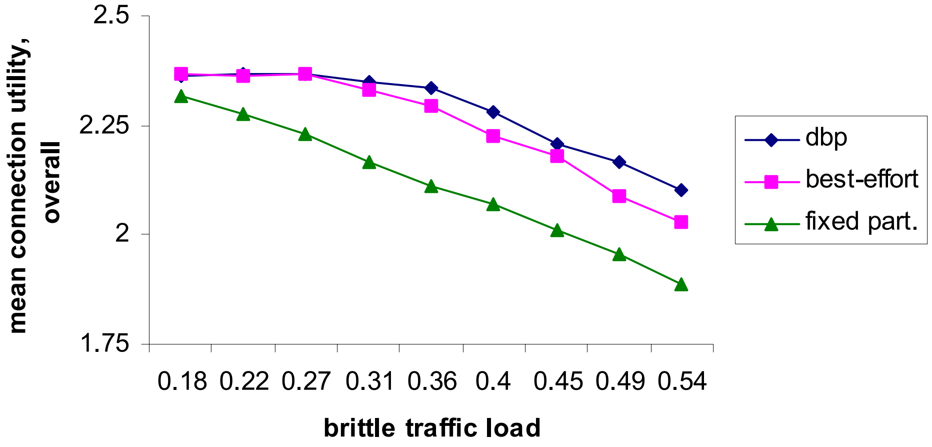

Figure 6.11. Comparison of the Mean Connection Utility, all Traffic Classes

Therefore, even though DBP limits the elasticity of elastic and stream traffic, when it comes to the measurement of the average end-users' satisfaction, modelled as described in this Thesis, DBP outperforms the best-effort concept of equal bandwidth allocation for all and no admission control. This is a very interesting conclusion. The reasons for such a result are numerous. Some of them will be analysed in the next three figures, which show the comparison for each of the three observed traffic classes.

It is interesting at this point to try to assess the margin between mean connection utilities for different bandwidth allocation schemes. The question is what does it mean to network operators, service providers, and finally, the end-users that the mean connection utility is approximately 0.05 larger when Dynamic Bandwidth Partitioning is used? Is such margin a valid reason for implementing a new network control, even if the control proves to be easy to implement?

The utility margin of 0.05 means that, on the average, end-users would be 0.05/2.25=2.2% *happier* if they are using the network that is implementing Dynamic Bandwidth Partitioning. This is a very small improvement, difficult to even register.

However, let us take a look at individual classes. Figure 6.12 shows that brittle traffic users will be on average 60% happier with using the network implementing the Dynamic Bandwidth Partitioning scheme. This means that two in three customers of a brittle application will not be satisfied with the application quality, and since it is a brittle, yes-or-no application, the user will not be able to use it at all.

This presents a very serious problem in performance. The introduction of additional control is necessary. The above analysis explains once more why it is important to observe both the overall mean connection utility, and to analyse each individual class.

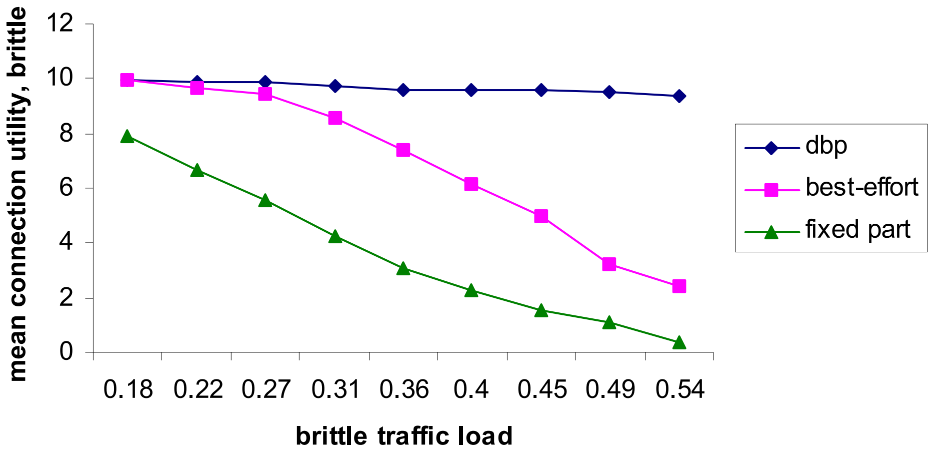

Figure 6.12. Comparison of the Mean Connection Utility, Brittle Traffic

The result presented on figure 6.12 is probably the most important for the Dynamic Bandwidth Partitioning scheme. The analysis of the brittle traffic is where we can see how powerful the new scheme is. DBP protects the brittle traffic by introducing three controls:

- admission control accepts a new brittle traffic flow only when there is enough bandwidth for it
- bandwidth partitioning protects brittle traffic flows from sudden bursts of elastic traffic
- prioritisation: the brittle traffic is defined to be more important than elastic traffic, by setting scaling parameters for the utility calculation

The outcome of these three controls can be seen in figure 6.12. In the area of high loads, only DBP is able to sustain high mean connection utility.

The best-effort scheme does not take into consideration the requirements of the brittle traffic flows. Since these flows are not elastic, it is essential that their bandwidth requirements are fulfilled. The best-effort scheme does not provide a means to do that. Best-effort scheme generates very low mean utility for large traffic loads, close to 4, which means that approximately only 40% of all brittle traffic flows receive the required bandwidth level for the whole duration.

Fixed bandwidth partitioning does provide the means by separating the capacity, but it does not react to the possible increase in the load of the brittle traffic. Even for small traffic loads fixed partitioning is not capable of generating a maximal mean utility, which means that even for these small loads there is a substantial rejection rate for the brittle traffic.

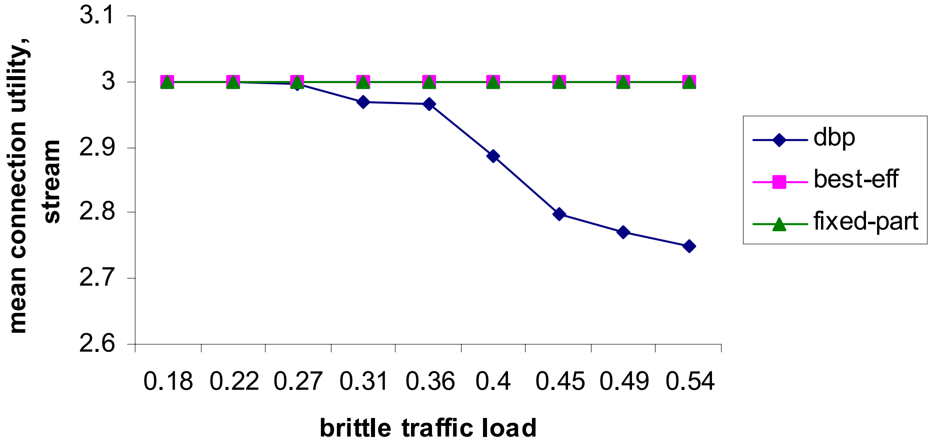

Figure 6.13. Comparison of Mean Connection Utility, Stream Traffic

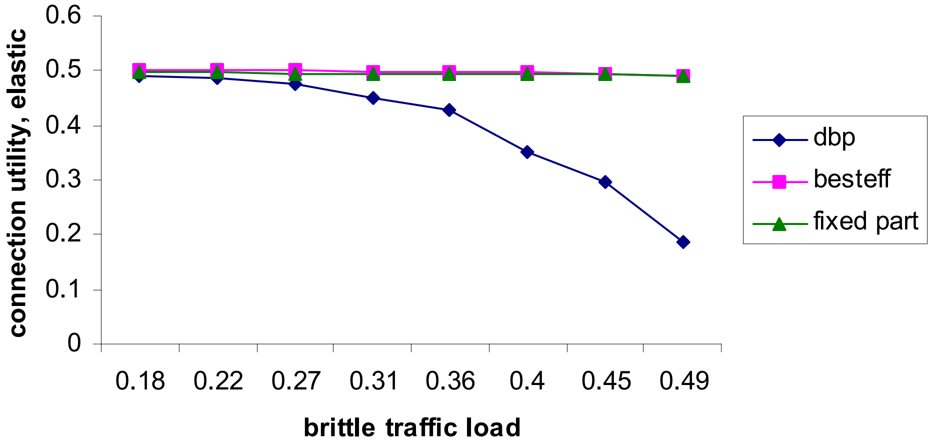

Figure 6.14. Comparison of Mean Connection Utility, Elastic Traffic

Figures 6.13 and 6.14 show that the conclusion is different when it comes to traffic flows with elastic features, belonging to either stream or elastic traffic class.

Stream traffic flows receive on average smaller utility in the Dynamic Bandwidth Partitioning scheme then in the two other schemes. Naturally, the utility margin of 0.25 for the stream traffic is not negligible, but can be considered as small. However, the same conclusion does not stand for the elastic traffic, which performs poorly with Dynamic Bandwidth Partitioning, having rather low mean connection utility for large traffic loads.

Best-effort and Fixed Bandwidth Partitioning schemes perform very well for the stream and elastic traffic. Best-effort scheme provides unlimited capacity for these traffic classes, enabling them to fully utilise its elasticity. The Fixed Bandwidth Partitioning concept provides a third of the link bandwidth to each of the traffic classes, without any interference, which results in excellent performance from both stream and elastic traffic.

Therefore, to draw the final conclusion about whether it is optimal or not to implement the Dynamic Bandwidth Partitioning is not an easy task. It depends on the number of different issues: the nature of the observed network, the traffic load that is put on the network, the application requirements, and the traffic prioritisation. All of these aspects must be taken under consideration, and after defining realistic utility functions and analysing the generated connection utility for all traffic classes, and overall, the conclusions can be made.

The results on figures 6.10-6.13 show the analysis of one traffic load pattern, where the traffic loads were approximately equal. The next section will analyse what happens when the traffic load from one of the traffic classes suddenly increases, i.e. when this traffic balance is disturbed.

6.5.5. Performance Comparison in the Case of the Sudden Traffic Fluctuations

The previous section analysed the case when the traffic loads for all observed traffic classes were approximately the same. It is a question whether this is ever the case in the real networks. That traffic pattern was used only as a starting point for the further analysis.

This section will provide the analysis of a large number of experiments that were performed as an attempt to cover other possible traffic patterns. It will be interesting to see how various bandwidth allocation concepts react to the sudden changes in the traffic load, and also to uneven distribution of the traffic loads.

As discussed above, the environment in which the network performance is observed is very important in reaching conclusions about the benefits generated by the Dynamic Bandwidth Partitioning scheme. For example, the network with heavy elastic load and very few users of the brittle traffic, does not require introduction of an additional scheme to protect only a minority of users.

On the other hand, there are many cases where it is proven that some control is needed. In this section one of such cases is experimented with, namely the case when there is a sudden fluctuation in the brittle traffic load.

In general, the way the experiments have been conducted is as follows. At simulation time $t = 120000$ sec the incoming rate of the traffic from one of the traffic classes increased 5 times. By doing this we want to explore how different bandwidth allocation concepts react to the sudden change. Furthermore, we want to see how they perform in the situation when the load of one of the traffic classes is much larger than the load of the other two.

The performance metric that was used is the mean connection utility. It was more interesting here to observe the comparison of the mean connection utility for individual classes, than to look at the overall comparison. The initial traffic load for this experiment was small, in order for the traffic burst to be more evident.

Figures 6.15-6.18 show what happens when the incoming rate for the brittle traffic increases 5 times. This load increase has substantial impact on the network state. Figure 6.15 shows that in this case the dynamics and the control introduced by the DBP scheme is fully utilised. DBP scheme is able to react to such a change in the network load. Naturally, the mean connection utility for the brittle traffic decreases, but this decrease is small compared with the other two schemes. The mean connection utility for the best-effort and fixed partitioning schemes decrease substantially, for fixed partitioning it even decreases below zero, which means that there are more rejected brittle traffic flows than accepted.

When it comes to stream and elastic traffic, the situation is different, and the conclusions we can make here are similar to the conclusions from the previous section. Both stream and elastic traffic suffer in the DBP scheme when there is a sudden increase in the brittle load. However, the level of decrease in the mean connection utility for stream traffic is only 0.2. The decrease for the elastic utility is substantial. On the other hand, best-effort scheme is able to sustain the high utility for the elastic traffic. This scheme provides equal conditions to all traffic classes, and the elasticity of the elastic traffic provides them with good 'surviving' skills in the case of a sudden burst in the traffic loads.

Best-Effort concept shows better results than the fixed partitioning concept, mostly because of its elasticity. Fixed partitioning was not able to provide maximal utility for the brittle traffic even before the load increase, and after the increase the mean utility drops even below 0, which means that there were more then 50% rejections.

For stream and elastic traffic, fixed partitioning performs better than best-effort. In the fixed partitioning scheme, the traffic classes are completely independent, so only the traffic class which increases its load suffers from that increase.

What is the effect on the overall utility? Figure 6.18 shows that Dynamic Bandwidth partitioning performs better overall. DBP is able to sustain the level of mean connection utility, while both best-effort and fixed partitioning decrease their utilities.

The sudden burst of brittle traffic brings huge problems to best-effort and fixed partitioning schemes, mainly concerning the utility of the brittle traffic. Dynamic Bandwidth Partitioning is able to sustain a high connection utility for the brittle traffic class, which is the main reason for outperforming the other two schemes.

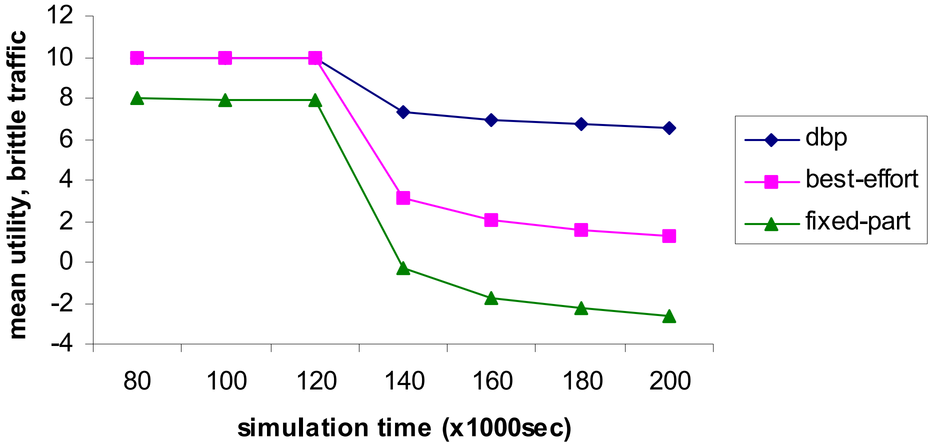

Figure 6.15. Mean Connection Utility for the Brittle Traffic when Brittle Burst Arrives

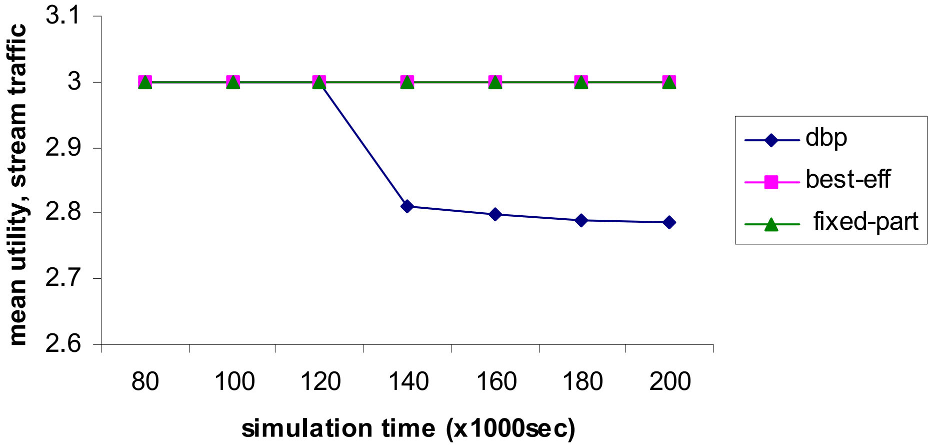

Figure 6.16. Mean Connection Utility for the Stream Traffic when Brittle Burst Arrives

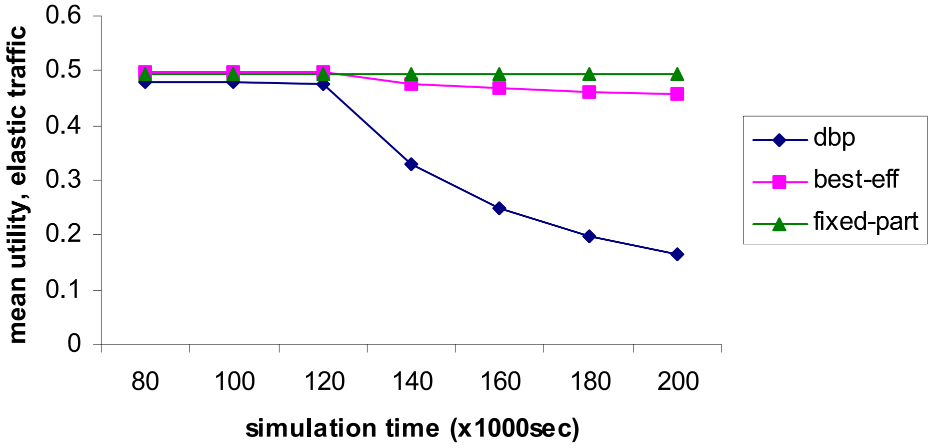

Figure 6.17. Mean Connection Utility for the Elastic Traffic when Brittle Burst Arrives

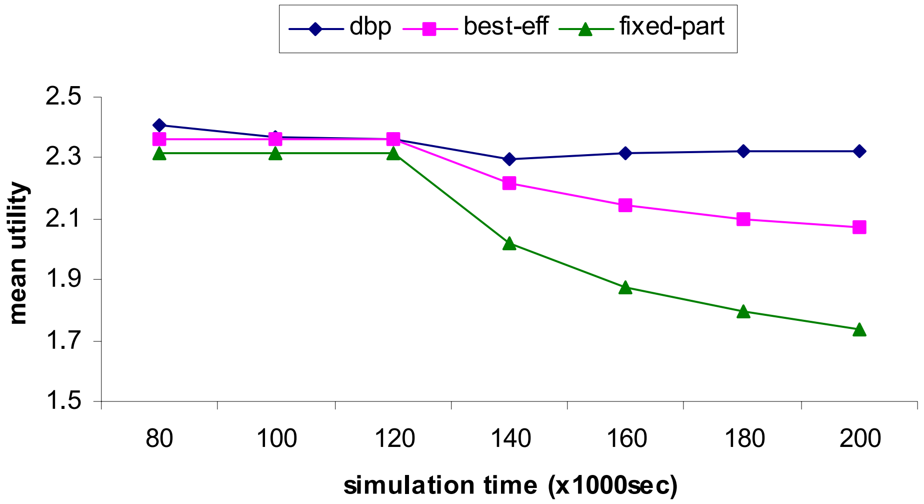

Figure 6.18. Mean Connection Utility Overall when Brittle Burst Arrives

Let us now consider the case when the incoming rate of *stream* traffic flows increases 5 times. Figures 6.19-6.22 show what happens to the mean connection utility for the three observed bandwidth allocation schemes. The best-effort scheme performs again badly for the brittle traffic, in the same way it performed after the load of the brittle traffic increased. Dynamic Bandwidth Partitioning survives the stream load increase very well, decreasing the mean connection utility for the brittle traffic just a fraction. Fixed bandwidth partitioning ignores the stream load increase when it comes to brittle traffic because the traffic classes are fully separated.

For the stream traffic, the impact is felt for both of the partitioning schemes, where DBP does not perform too badly, decreasing its mean connection utility for only 0.05.

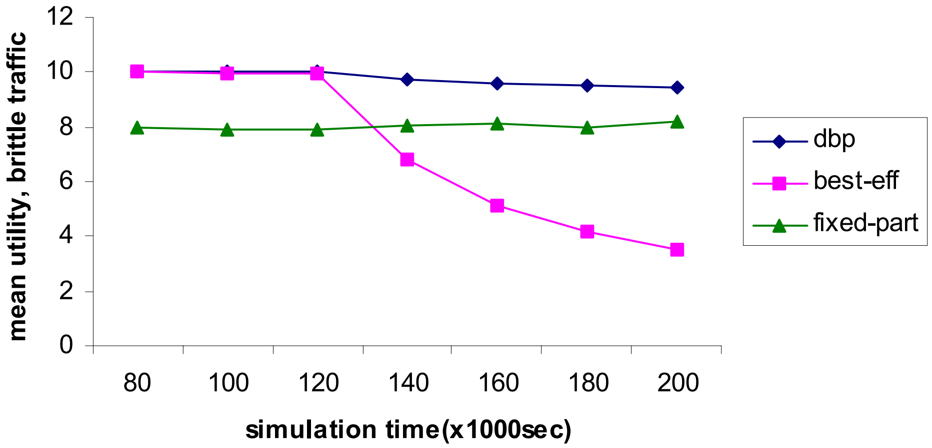

Figure 6.19 Mean Connection Utility for the Brittle Traffic when Stream Burst Arrives

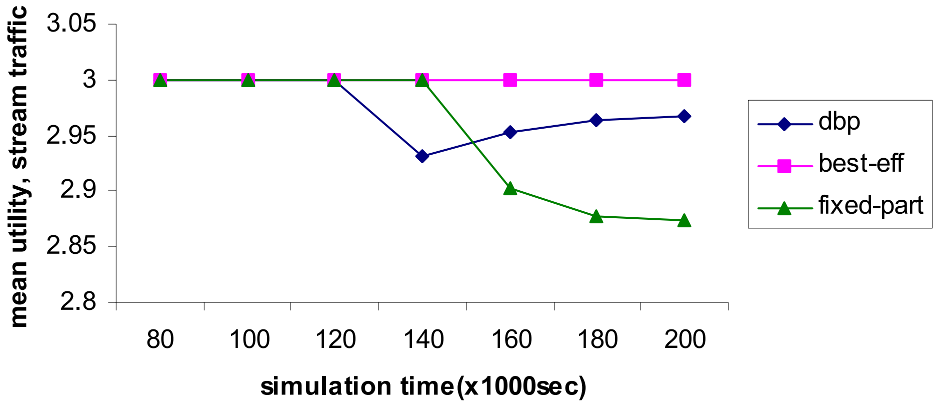

Figure 6.20. Mean Connection Utility for the Stream Traffic when Stream Burst Arrives

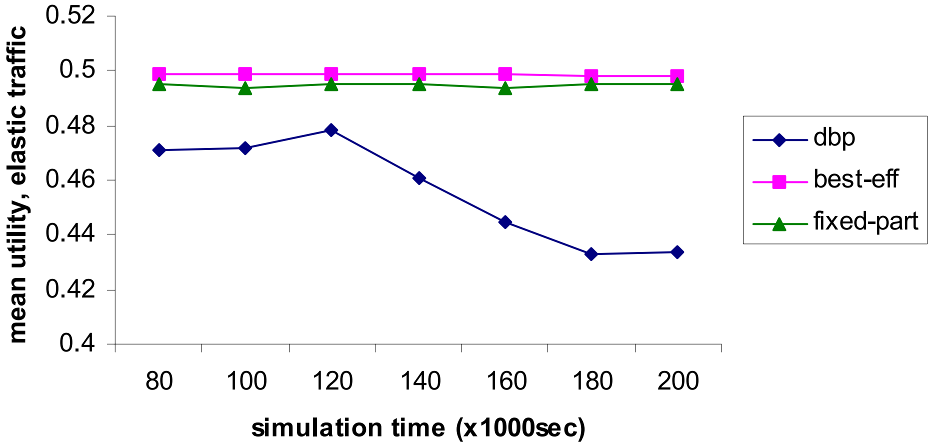

Figure 6.21. Mean Connection Utility for the Elastic Traffic when Stream Burst Arrives

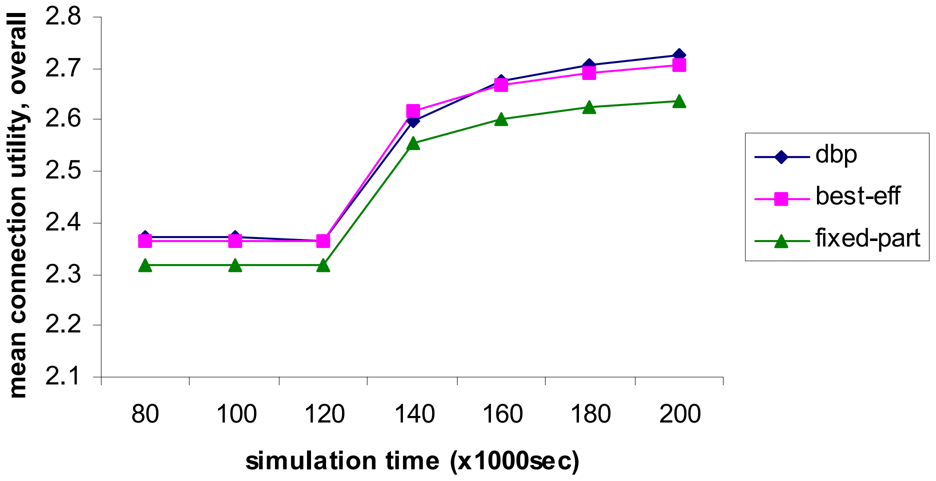

Figure 6.22. Mean Connection Utility Overall when Stream Burst Arrives

Finally, for elastic traffic, it is DBP again that performs poorly, due to the reasons explained already in this Chapter. However, the decrease in utility is not bigger than 0.05, which does not mean much for the elastic applications.

Interesting results are shown in figure 6.22. There is no benefit from using the Dynamic Bandwidth Partitioning scheme when it comes to overall mean connection utility. The interesting point here is that the overall utility *increases* for each of the bandwidth allocation concepts. This is due to the fact that there are many more stream traffic flows that contribute to the overall utility and increase it. We can see that the level of the overall utility is 2.3-2.8, meaning that a 'successful' stream traffic flow, with the connection utility equal to 3, would increase the overall utility. This was not the case with the increase in brittle flows (see figure 6.18), because that burst decreased substantially the mean utility for the brittle traffic

In summary, the sudden load increase for stream traffic creates problems to the best-effort scheme. Dynamic Bandwidth Partitioning survives very well, and fixed partitioning performs poorly only for the stream traffic. We know from previous analysis that DBP performs slightly worse than best-effort for the stream flows in the environment of high traffic loads (see figure 6.16). However, the results shown here prove that when it comes to sudden bursts in the stream traffic, or any kind of dynamic change in the traffic load, or in the case when stream traffic load is dominant over brittle and elastic, DBP is a better overall choice.

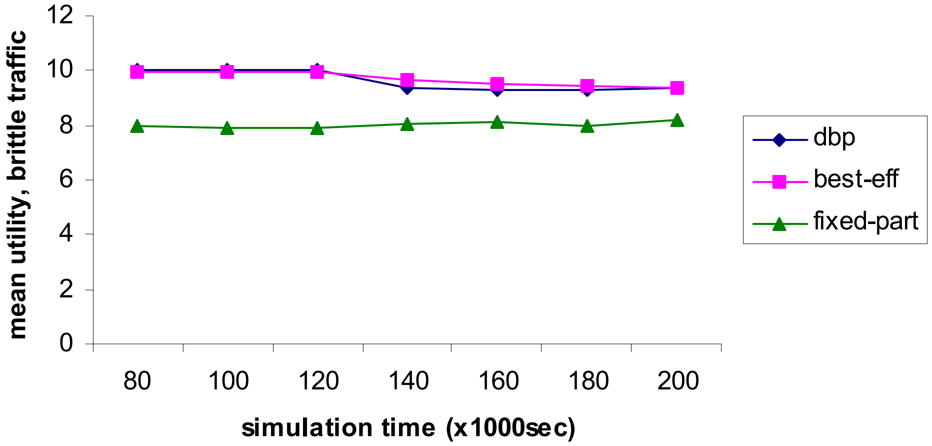

Figure 6.23. Mean Connection Utility for the Brittle Traffic when Elastic Burst Arrives

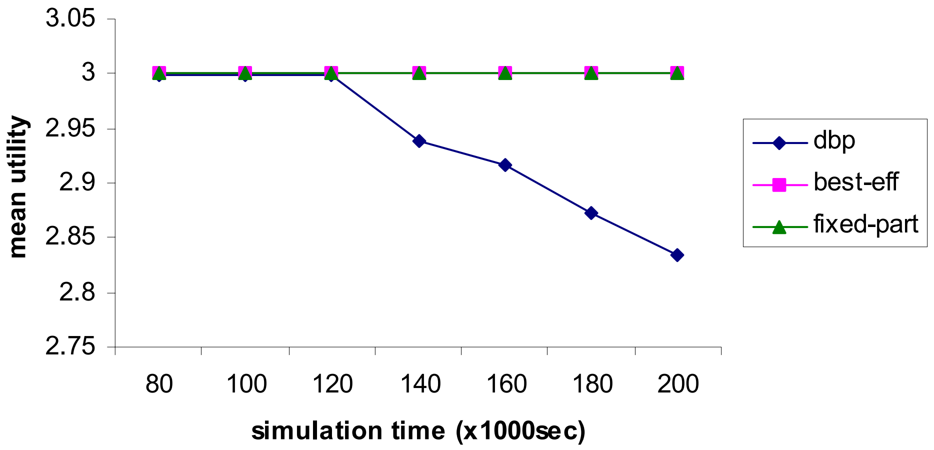

Figure 6.24. Mean Connection Utility for the Stream Traffic when Elastic Burst Arrives

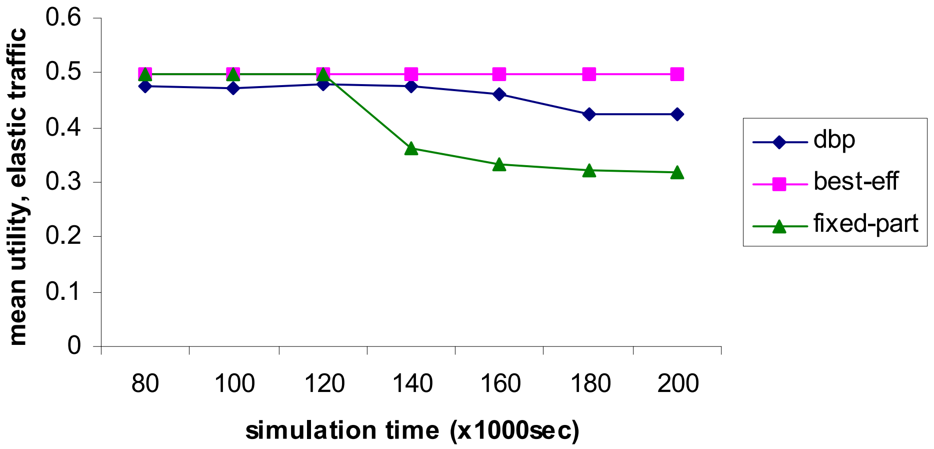

Figure 6.25. Mean Connection Utility for the Elastic Traffic when Elastic Burst Arrives

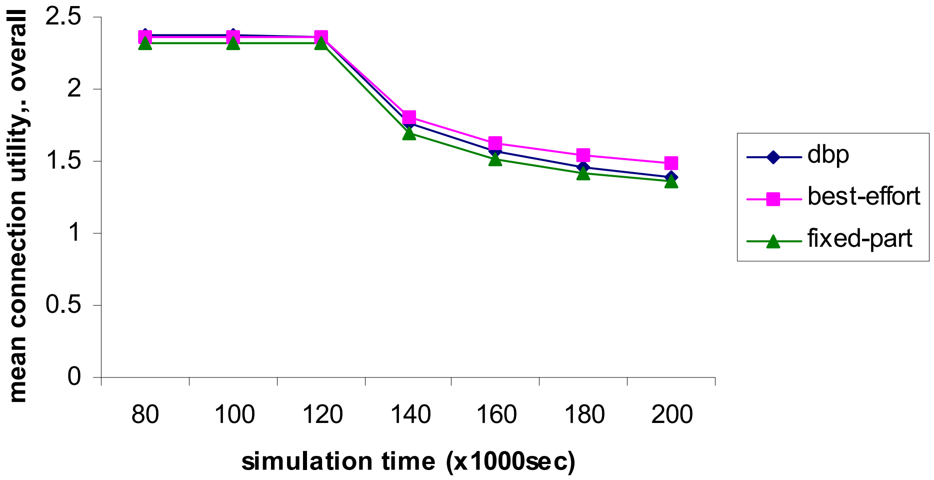

Figure 6.26. Mean Connection Utility Overall when Elastic Burst Arrives

Figures 6.23-6.26 show the impact of the increase in *elastic* traffic load. As we can see from the figures, this impact is smaller than the impact of the load increase for brittle and stream traffic. The impact on the mean connection utility for all three bandwidth allocation schemes is negligible, and so it is for the stream traffic. Even the two critical relations, brittle traffic for the best-effort scheme and elastic traffic for the Dynamic Bandwidth Partitioning scheme, are not affected. It is only fixed partitioning for elastic traffic that shows bad performance, which is due to the lack of dynamics and the high elastic traffic load.

We can see on figure 6.26 that all three bandwidth allocation schemes change the overall mean connection utility in the same way. The overall utility decrease is mainly due to the increased in the number of elastic traffic flows.

These results show that Dynamic Bandwidth Partitioning is able to cope with the increased elastic load. The effect of the large elastic load when the DBP scheme is used can be seen only if there is an increase in all three traffic loads. Then the elastic traffic is the one that suffers most, as we can see in figure 6.17.

6.5.6. Performance Comparison for Different Link Capacities

The implementation issues for Dynamic Bandwidth Partitioning are almost as important as the measured performance of the scheme. It is of no use to have the scheme with excellent performance, if the scheme is too expensive and difficult to use, or if it requires complicated signalling which creates excessive overhead, thus unacceptably increasing the congestion in the network.

In order to determine the optimal environment for the Dynamic Bandwidth Partitioning scheme, an experiment has been conducted in which the traffic loads are kept constant, while the link capacity is changed.

It is important to see how different bandwidth allocation concepts react to different link capacities. Different link capacities actually mean different networks, since smaller link capacities correspond to the networks with limited bandwidth (such as access networks), while large link capacities correspond to the core networks.

Naturally, it is expected that the access networks will require more control in the bandwidth allocation, since their resources are scarce, and it is more likely that higher-priority traffic will suffer due to insufficient bandwidth. In the network with larger capacity there is usually enough bandwidth for the entire traffic, and the introduction of additional control is not necessary.

Another interesting point to note is that the smaller link capacities actually correspond to larger network loads, and therefore the results presented here are very similar to the results presented earlier in this chapter. Nevertheless, it was important for us to conduct these experiments in order to reach valid conclusions.

Figure 6.27 shows the results of the experiment described above. A ‘middle’ range of link capacities was analysed, between 65Mbit/s and 140Mbit/s. The conclusions for the capacities smaller then 65Mbit/s and larger than 140Mbit/s are obvious from figure 6.27. In other experiments presented in this Chapter the link capacity used was 100 Mbit/s.

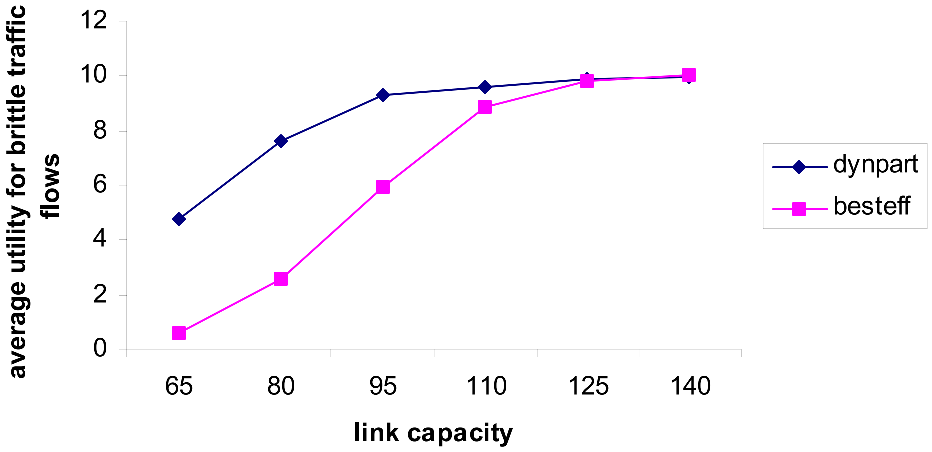

Figure 6.27. Comparison of Mean Connection Utility for Brittle Traffic for Various Link Capacities

Figure 6.27 compares the behaviour of Dynamic Bandwidth Partitioning and the Best-effort scheme. The comparison was based on mean connection utility for brittle traffic flows. As expected, the figure shows that if the link capacity is small, DBP is superior to the best-effort concept when it comes to brittle traffic. For small link capacities best-effort shows very poor results, almost completely disabling brittle users from getting respectable performance.

Elastic flows suffer with Dynamic Bandwidth Partitioning, but not as much as brittle flows suffer with the best-effort concept. Figure 6.28 shows the comparison of the mean connection utility for the elastic traffic. The expected result is that a best-effort scheme generates higher connection utility for both smaller and larger link capacities.

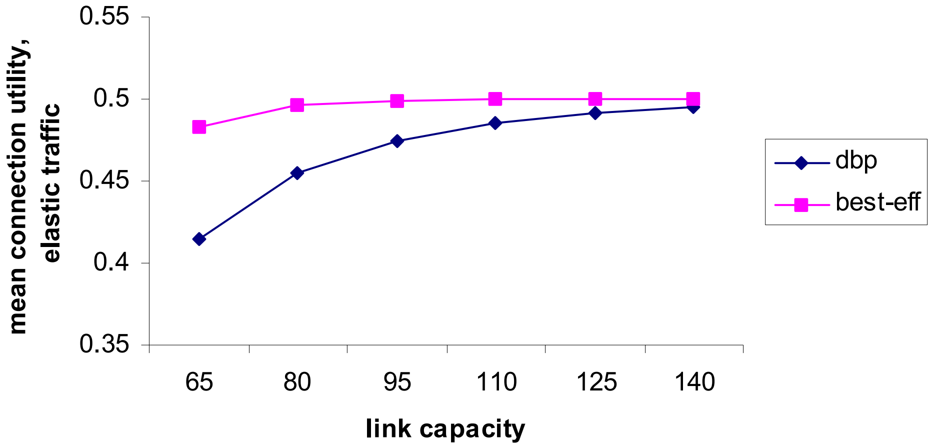

Figure 6.28. Comparison of Mean Connection Utility for Elastic Traffic for Various Link Capacities

An important conclusion from the simulation results presented here is as follows: we can see from figures 6.27 and 6.28 that DBP outperforms the best-effort scheme when it comes to the brittle traffic for smaller link capacities. This proves once more the general conclusion about partitioning schemes performing better in the areas of high traffic loads. Therefore, the networks that may find optimal the use of a Dynamic Bandwidth Partitioning scheme need to have two important features: *limited (small) capacity* and *traffic diversity*.

Bandwidth partitioning is unlikely to be the optimal resource allocation scheme for the core IP network, where high-speed optical routers are likely to provide enough bandwidth so that simpler allocation mechanisms can be used. However, in the networks with limited capacity, such as the access networks, this type of bandwidth allocation can prove to be very useful. Access links, even after using sophisticated technologies such as ADSL, still present a bottleneck for the network. In the networks with limited resources, we need a resource allocation scheme that can at the same time provide service for as many as possible traffic connections, and give appropriate performance to those connections, maximizing the connection utility and the revenue at the same time.

6.5.7. Performance Comparison for Different Scaling Parameters

The scaling parameters, denoted ω_i in Chapter 4, are the parameters used for prioritisation of the traffic classes. They are used to emphasise the fact that real-time, brittle traffic should be more important in the decision-making for the partitioning updates then the elastic traffic. The larger the ratio between the scaling parameters is, the more regularly will the brittle traffic trigger the partitioning updates, while a large utility decrease for elastic traffic would be necessary to trigger the partitioning update.

Furthermore, scaling parameters present a very important control that is given to the network operators. It is possible, based on what the network providers want to do with the network, to change the scaling parameters and in that way make the traffic differentiation more or less important. Therefore, depending on the kind of network that needs to be designed, different scaling parameters can be used to define the relations between traffic classes.

An interesting question that arises here is the question of pricing for individual services and traffic classes. The scaling parameters should be in some relation with the prices that are put on the traffic. The same point is valid for the connection utility, which as a measure of end-users' satisfaction needs to include the pricing mechanism. However, detailed discussion about the pricing for individual network applications and traffic flows is out of the scope of this Thesis.

In this section the impact of different scaling schemes is analysed. In order to assess this impact, a new metric is introduced, the *relative utility ratio* of mean connection utilities for different bandwidth allocation schemes. We denote this ratio δ.

The mean connection utility is calculated from

$$u_{mean} = \frac{\sum \omega_i n_i u_{ji}}{\sum n_i} \qquad (6.5)$$

We define the relative utility ratio δ as

$$\delta = \frac{u_{mean}(dynpart) - u_{mean}(besteff)}{u_{mean}(dynpart)} \qquad (6.6)$$

where $u_{mean}(dynpart)$ is the mean connection utility when the DBP scheme is used, and $u_{mean}(besteff)$ is the mean connection utility when the best-effort scheme is used.

In the experiment, the relative utility ratio δ is measured for four different scaling schemes, and for different traffic load patterns. The traffic load patterns are the same as in the previous experiments. The scaling schemes range from equal prioritisation, $\omega_B = \omega_S = \omega_E = 1$, to the clear difference in the priority, when the maximal utility of a brittle traffic flow is valued 20 times more then the maximal utility of an elastic traffic flow, $\omega_B = 10$, $\omega_S = 3$, $\omega_E = 0.5$. The objective of this experiment is to show that the choice of the scaling parameters is very important.

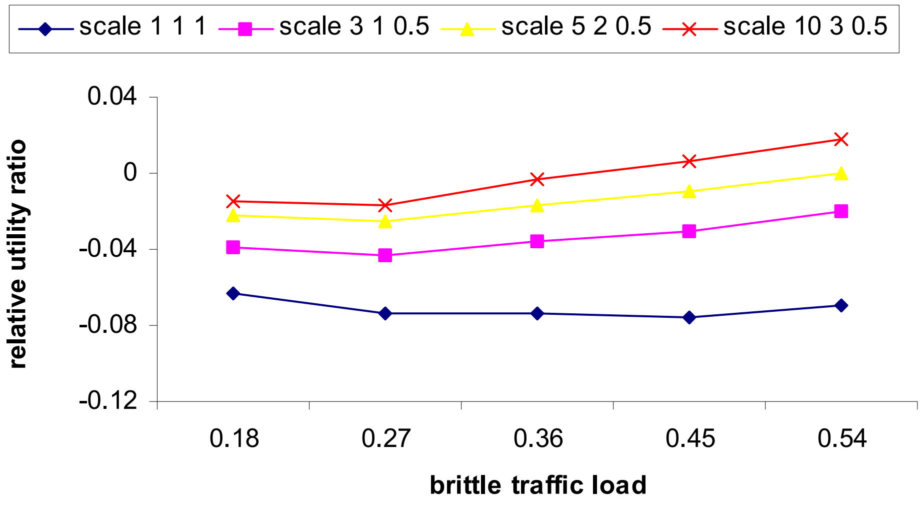

Figure 6.29. Impact of the Scaling Parameters

Figure 6.29 shows the results of the experiment. It is clear that the larger the difference between the scaling parameters is, the greater is the benefit from the implementation of Dynamic Bandwidth Partitioning. The relative utility ratio increases with the load increase for all four scaling schemes, showing once more that the need for bandwidth allocation control rises with increasing traffic load.

The interesting conclusion from this result is that the comparison between DBP scheme and the best-effort concept depends heavily on the chosen scaling constants. For the first three combinations of scaling parameters, the relative ratio is negative, meaning that the best-effort concept generated higher mean utility than the DBP scheme.

The conclusion is that, depending on how much more are users of the brittle traffic important to the network operators DBP scheme will be more efficient and optimal to use.

This complies with the previous conclusion that the real efficiency of the DBP concept depends heavily on the type, and the capacity of the observed network, as well as on the average traffic load in the network.

6.5.8. Analysis of the Blocking Rate

This section reports an interesting experiment that examines the negative impact of bandwidth allocation schemes on high-priority brittle traffic. It has been mentioned many times before that one of the main differences between Dynamic Bandwidth Partitioning and Best-Effort schemes is in admission control. DBP introduces a strict admission control for the brittle traffic, where a new brittle flow is not allowed to enter the network if there is not enough available bandwidth $b_{b\min}$ for it. On the other hand, a best-effort scheme does not have such a control, which means that brittle traffic flows are always accepted in the network, whatever the network state is. This introduces unwanted effects in the network, such as when too many brittle flows are on the network link at the same time, producing congestion and decreasing the bandwidth levels beyond the required minimal level $b_{b\min}$. Once that happens, the end-users are unable to use the applications. The utility of these applications then becomes 0.

There is an obvious trade-off here. On one hand, there are applications whose requests are being rejected in the Dynamic Bandwidth Partitioning scheme. On the other hand, there are applications that are accepted in the network in the best-effort scheme, but the network performance is poor and they become useless to the end-users.

The question is which of these two undesirable effects is worse.

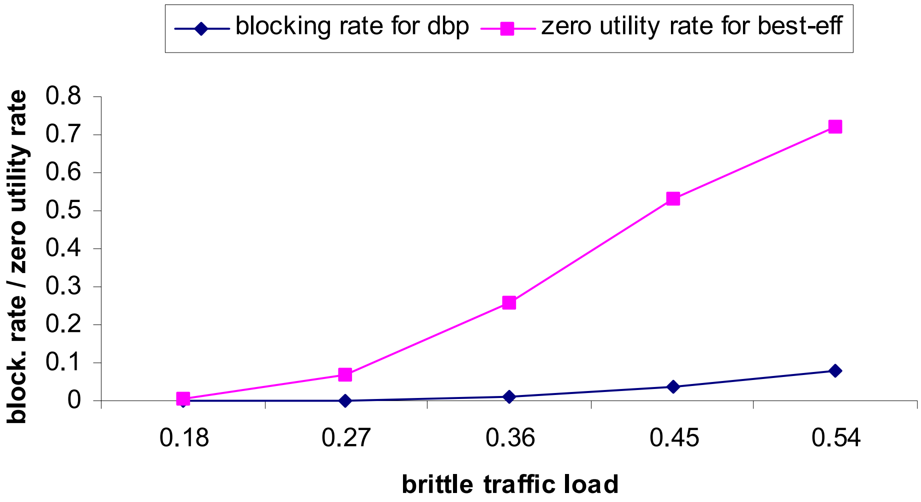

Figure 6.30 Comparison of the Negative Impact on the Brittle Traffic

The blue line on figure 6.30 shows the blocking rate for the brittle traffic when the DBP scheme is introduced. The blocking rate for the brittle traffic is not small, especially for very high brittle traffic loads. However, for the brittle traffic loads of about 0.4 the blocking rate is only 3-5%. It is only for very high traffic loads that the blocking probability becomes very high. The traffic load in this experiment is like in Table 6.8.

On the other hand, the pink line on figure 6.30 shows the percentage of brittle traffic flows that are not allocated the required bandwidth for the whole duration of the flow. It is obvious from the figure 6.30 that even moderate traffic loads for brittle traffic (0.3-0.4) generate very high rate of such zero-utility brittle connections.

The advantage of the introduction of Dynamic Bandwidth Partitioning in this case is obvious. This result is another indication of the efficiency of the DBP scheme when it comes to QoS-aware, brittle traffic connections.

6.5.9. Summary

This section presented the performance comparison of Dynamic Bandwidth Partitioning and two other bandwidth allocation schemes, best-effort and fixed bandwidth partitioning. The best-effort scheme is the scheme where the traffic is fully aggregated, and all traffic flows are treated in the same way, regardless of their performance requirements. In fixed bandwidth partitioning, the available bandwidth is partitioned between traffic classes, but there is no dynamics in the partitioning, i.e. the bandwidth partitioning parameters are fixed.

The three schemes have been compared on the basis of three measures: mean connection utility, mean link utilisation and mean traffic flow duration for elastic traffic.

The comparison of the mean link utilisation showed that best-effort scheme outperforms the two bandwidth partitioning schemes when it comes to the amount of bandwidth used on the average. This result is expected, since both of the bandwidth partitioning schemes limit the elasticity of elastic and stream traffic flows, thus decreasing the overall used bandwidth. Dynamic Bandwidth Partitioning performs better then fixed partitioning, because of the dynamics of the partitioning algorithm. The margin between the best-effort and Dynamic Bandwidth Partitioning is fairly small, around 5% for the high traffic load.

The margin is substantially higher, however, when it comes to the comparison of the mean flow duration for the elastic traffic. Both best-effort and fixed bandwidth partitioning perform well when it comes to this measure, but DBP performs poorly, especially for high traffic loads. This is due to the traffic prioritisation that is introduced in DBP. The elastic traffic has the lowest priority and suffers in the area of high brittle and stream loads.

The performance of the elastic traffic is the biggest problem of the DBP scheme. However, the traffic prioritisation is the inherent feature of the scheme, and the trade-off between the assured performance guarantees for the real-time traffic classes (brittle and stream) and the poor performance of the elastic traffic presents a necessary drawback of the scheme. The problem is to assess whether this drawback has been successfully compensated through the performance improvement for the high-priority traffic. Connection utility is a new metric that was introduced for this purpose.

The comparison of the mean connection utility for all traffic classes shows that DBP *does* provide better overall performance when all traffic classes are considered. Figure 6.11 shows that even though the margin is not big, the dynamics and admission control introduced by the DBP scheme result in higher mean connection utility. Naturally, this result depends on the traffic loads that were presented to the network. Further analysis of individual traffic classes shows that DBP performs very well for the brittle traffic. It is capable, even for high traffic load, to sustain maximal connection utility. Both best-effort and fixed bandwidth partitioning perform poorly for the brittle traffic, mainly because the lack of the traffic prioritisation makes all traffic classes equal and provides poor performance for the high-priority brittle traffic. On the other hand, when it comes to the elastic traffic, DBP performs worse than the other two schemes. DBP was designed to provide performance guarantees to the higher-priority traffic, and the poorer performance of the lower-priority elastic traffic is a natural consequence. By measuring the overall mean connection utility and comparing it for the three bandwidth allocation concepts, it is possible to assess the extent of the decrease in the performance for the elastic traffic when DBP is used. Figure 6.11 shows that this drawback of our scheme is fully compensated by the provision of performance guarantees to higher-priority traffic classes.

Furthermore, this section studied the performance of the three schemes when the traffic is not distributed evenly. The experiments were done for the sudden fluctuation of the traffic from each of the traffic classes. The results showed that DBP handled the sudden burst in brittle traffic flows very well, compared to the poor performance of the other two schemes. For the bursts of elastic and stream traffic flows, DBP showed very similar performance to the other two schemes.

The other experiments in this section included the study of the scheme performance for variable link capacity and the analysis of the impact of the scaling parameters on the scheme performance. The conclusion of the first experiment is that DBP outperforms best-effort for smaller link capacities. This means that there are two important features necessary for the network to consider using the DBP scheme: limited capacity and traffic diversity. Networks with large capacity do not require new controls since they have enough bandwidth to provide good performance for the entire traffic even with using simple schemes. On the other hand, the networks with limited capacity need efficient controls to be able to maximise the performance and the number of satisfied end-users. Traffic diversity is also important, because different traffic classes need to be treated differently in the network. If the network is serving only one type of applications, there is no need for a control such as the one presented in this Thesis.

The traffic prioritisation is another important issue. In the experiment in section 6.5.7 different scaling mechanisms were used in order to see how does the scaling mechanism influence the final network performance. The result shows that DBP provides better overall performance when the difference in priority is bigger. The reason for this is that DBP provides performance guarantees for the brittle traffic on the expense of the elastic traffic. Therefore, the more important brittle traffic is, the overall benefit of the scheme is greater.

The overall conclusion of the experiments presented in this Chapter is that the introduction of a DBP concept can prove to be efficient. The necessity for such a concept depends heavily on the observed network, on the traffic and user profile. The networks with small capacity and diverse set of user-applications will find such a scheme very useful. This Chapter explained the controls network operators would have in the DBP scheme, where by setting different shaping rules and bandwidth requirements they would be able to define different utility functions and thus influence the bandwidth allocation. Furthermore, it has been shown that the network with large fluctuations in traffic, or the network with very large load of the highest priority traffic class needs such a scheme in order to maximise its performance.

The next Chapter will further discuss the results presented here and underline the important issues which could prove interesting for the future research on the implementation of the Dynamic Bandwidth Partitioning concept.

CHAPTER 7

DISCUSSION AND CONCLUSION

7.1. Discussion

The research presented in this Thesis addressed the problem of bandwidth allocation in the multi-class integrated TCP/IP-based Internet. The main constraint of the current Internet architecture is its inability to successfully integrate video, voice and data traffic while providing performance guarantees to the traffic connections that require these guarantees. In other words, the current Internet is unable to provide quality of service (QoS) guarantees to sophisticated real-time user-applications.

The objective of our research was the full formulation *of Dynamic Bandwidth Partitioning*, a new bandwidth allocation scheme which successfully addresses the QoS problem and creates a dynamic environment in which the bandwidth is allocated with requested guarantees in a way which maximises the mean level of end-users' satisfaction.

Dynamic Bandwidth Partitioning is a novel approach to the integration of diverse classes of Internet traffic. It involves the introduction of the bandwidth partitioning concept, in which the link capacity is strictly partitioned into fractions, making the forwarding of different traffic classes fully independent. This concept enables the network to guarantee certain performance level to the traffic classes that require it.

In the traffic model that has been used in the Thesis, the Internet traffic has been differentiated into three main classes, which are defined as brittle, stream and elastic. Brittle and stream traffic originates from real-time network applications. Brittle applications require constant bandwidth level while on the network, while stream applications require only the minimal bandwidth level, after which they are able to adapt to the bandwidth the network allocates them. The elastic traffic originates from data transfer applications and does not require any performance guarantees.

The bandwidth allocation in the Dynamic Bandwidth Partitioning scheme is optimised to increase the end-users' satisfaction with the network service. The way bandwidth is partitioned is constantly being updated based on the changes in the normalised end-user *utility.* This is being done according to the *partitioning algorithm*, which is defined in detail in Chapter 4.

Using utility as the measure of the network performance presents a novel approach in performance evaluation. The rationale behind this is that the value of a traffic connection, or the utility that the end-users have received form the network, is closely related to the network performance the traffic connection gets while being transported through the network. Utility approximates the level of end-user satisfaction, rating it on the scale from 0 to 1, based on the bandwidth level the traffic connection was allocated.

Utility is modelled by using *utility functions*. It is very difficult to accurately define the utility functions, mainly because each individual user of an Internet application has its own utility function. This is the reason why a single utility function is used for each of the traffic classes. Utility functions are analysed in detail in this Thesis. Different shapes of utility functions have the impact on the final performance of Dynamic Bandwidth Partitioning.

End-users' utility is used not only as the optimisation parameter in the partitioning algorithm but also as the evaluation metric used for assessing the benefits of the Dynamic Bandwidth Partitioning scheme. A novel metric, *connection utility,* is defined here for this purpose. Connection utility can be simply defined as the average utility level a traffic flow acquired while on the network. It is calculated after the traffic flow terminates.

This Thesis evaluates the Dynamic Bandwidth Partitioning scheme in detail. In order to generate accurate conclusions about the scheme performance, a number of extensive simulations have been performed. A specially developed event-driven simulator, written in C, was used for the simulations.

The first part of the analysis was the analysis of the scheme as it was defined in Chapter 4. A number of important parameters have been experimented with, in order to define the scheme more accurately, and to explain why was the scheme defined in the way it was defined in Chapter 4. Although the conclusions of this analysis are given in more detail in Chapter 6, here the summary of the conclusions is given.

First of all, the impact of the *shape* of utility functions was analysed. For elastic traffic, the simulation results showed that the average traffic flow duration increased with the decrease in the shaping parameter a_e. The increased parameter a_e means the utility function is less rigid. This means that the more rigid requirement elastic traffic flows put on the network, the performance they get will be better. Similar conclusion came after the analysis of the impact of different parameters for the stream utility function. The final conclusion of this analysis is that the way utility functions are defined not only defines the performance requirements of different traffic classes, but it also influences the dynamics of the scheme, optimising the bandwidth allocation according to the performance requirements. This is a very positive feature of the Dynamic Bandwidth Partitioning scheme.

When it comes to the decrease parameter ε, it was not easy to draw a single conclusion. The partitioning algorithm uses additive increase, additive decrease linear control to change the partitioning parameters. The decrease parameter ε is the step value for the linear control algorithm. The parameter ε defines the level of *dynamics* of the partitioning algorithm. Different traffic classes react differently to the changes in the partitioning dynamics. While brittle traffic prefers medium values for ε, stream traffic prefers critical values, large or small, and elastic traffic finds large values for ε optimal. The reason for this is simple – elastic traffic prefers large fluctuations in the bandwidth allocation, because in that environment elasticity of these traffic flows is fully utilised. On the other hand, the brittle traffic does not have elastic features; it prefers a strictly controlled network. The optimal decrease parameter ε, therefore, needs to be chosen depending on the prioritisation, traffic loads, and the observed network.

Another important issue was the issue of the *update interval* in the partitioning algorithm. The best choice for the update interval is the medium value, since both too small and too large values produce lower mean connection utility. Large update interval cannot increase the performance of the elastic traffic, since this type of traffic is very dynamic, and prefers smaller update intervals. On the other hand, the small time interval for the update means that the updates are happening very frequently, often more frequently then the changes in the network state. Similar results exist for the stream and brittle traffic, in each of the cases it is the medium update interval that brings better performance.

The second major analysis is the comparison of the performance of Dynamic Bandwidth Partitioning scheme and the two other bandwidth allocation concepts, the full aggregation of traffic ('best-effort' concept), and the fixed bandwidth partitioning concept. Our scheme is compared with these two schemes on the basis of mean connection utility and on the basis of conventional metrics, such as mean link utilisation, and mean traffic flow duration (file transfer time). Other conventional metrics, such as end-to-end delay, delay variation and packet loss have not been used, since the network model consisted of a single link. Furthermore, the connection-level simulation has been performed, and all the packet-level experiments (including the measurement of the delay variation and the packet loss) have been omitted.

The main conclusion from these experiments is that there are traffic and network environments in which Dynamic Bandwidth Partitioning outperforms the other two schemes. These environments are usually the ones with high network load (or low link capacity), especially high load of brittle traffic. The network without dynamic control finds it very difficult to serve the high brittle traffic load.

The problem of assessing the level of the performance margin between different schemes is very important. In a number of figures in Chapter 6, the comparison of the mean connection utility can be observed. Assessing the true meaning of a certain utility margin is a crucial problem. For example, there are many cases when the utility difference can prove to be small, and where there is no point in introducing the network control (however simple it is), just in order to achieve a couple of percent increase in the performance. Every single network is a story for itself: the need for strict control must be assessed, and appropriate decisions need to be arranged accordingly. That is why the implementation simplicity is of absolute priority.

When it comes to the conventional metrics, the mean utilisation of the link bandwidth for the best-effort concept is higher then in the case of bandwidth partitioning, both fixed and dynamic. When the link bandwidth is not partitioned, the traffic is not restricted to using only a limited portion of the link bandwidth, so the elasticity of the elastic and stream traffic can extend to the maximum.

On the other hand, Dynamic Bandwidth Partitioning is created to provide bandwidth guarantees, i.e. to protect the applications that require such guarantees. Since we are considering the constant link capacity, elastic traffic must suffer from the introduction of the new control mechanism, in order for the appropriate level of capacity to be given to the users of brittle applications. This produces the increase in the mean traffic flow duration for elastic traffic in Dynamic Bandwidth Partitioning. This increase is rather large.

Therefore, both mean link utilisation and the mean traffic flow duration show the advantages of an unlimited bandwidth allocation scheme with no admission control and no network control. However, this scheme is unable to provide guarantees to the applications that require these guarantees. How can this be assessed and compared with the positive aspects of the *'accept all'* best-effort policy? The need for a universal metric that provides full information of the network performance is obvious. Different traffic classes require different network performance, and it is important to include new ideas in the evaluation of the network performance.

This is why the **connection utility** was defined. Simulation results show that, as the traffic load increases, Dynamic Bandwidth Partitioning shows slightly better performance then the best-effort scheme. This becomes more obvious for higher traffic loads. Fixed partitioning performs poorly, generating lower mean connection utility for both small and large traffic loads. Therefore, even though Dynamic Bandwidth Partitioning limits the elasticity of elastic and stream traffic, when it comes to the measurement of the average end-users' satisfaction, modelled as described in this Thesis through connection utility, it outperforms the best-effort concept. This is a very interesting conclusion. The reasons for such a result are numerous.

Firstly, Dynamic Bandwidth Partitioning protects the brittle traffic by introducing three controls:

- admission control accepts a new brittle traffic flow only when there is enough bandwidth for it
- bandwidth partitioning protects brittle traffic flows from sudden bursts of elastic traffic
- prioritisation: the brittle traffic is defined to be more important then elastic traffic, by setting scaling parameters for the utility calculation

The impact of these controls is obvious when we measure the mean utility generated for the brittle traffic connections. Dynamic Bandwidth Partitioning is clearly superior when it comes to the brittle traffic. The traffic aggregation concept is unable to provide the first two controls, leaving numerous brittle connections with less bandwidth than they require (which makes those applications useless for the end-users). On the other hand, the fixed bandwidth partitioning concept cannot adapt to the network load changes. As soon as the brittle traffic load becomes too large for the capacity that is reserved for it, the admission control starts to reject the incoming brittle traffic flows and the mean connection utility decreases rapidly.

Since we analyse the three different traffic classes, it was interesting to see how a sudden fluctuation of the traffic from one of the traffic classes influences the network performance. When the incoming rate for the brittle traffic increased 5 times, the dynamics and the control introduced by the Dynamic Bandwidth Partitioning scheme was fully utilised. The network was able to react on such a change in the network load. Naturally, the mean connection utility for the brittle traffic decreased, but this decrease was small compared with the other two schemes. However, both stream and elastic traffic suffer in the DBP scheme when there is a sudden increase in the brittle load. A sudden increase in both stream and elastic load results in similar behaviour of all three bandwidth allocation schemes, with fixed partitioning providing the worst performance.

There were a number of issues, including the definition of the utility functions, the implementation analysis, the use of MPLS virtual paths, and the definition of Hierarchical Dynamic Bandwidth Partitioning, that were done in order to give a full description of the scheme. All the non-essential issues, such as the full analysis of the end-to-end implementation, and the implementation of utility-based admission control and routing, have been omitted in the Thesis because the goal was to explain the new concept and to analyse in many detail the implementation of the scheme on a single network link. These issues are discussed in the next section, where the guidelines for the future research on Dynamic Bandwidth Partitioning will be given.

7.2. Future Work

This Thesis presented a comprehensive definition of a new scheme and can serve as a good basis for a number of future research issues that arise from defining a new bandwidth management scheme. Some of those issues will be analysed in this section.

The first interesting issue that will be analysed here is the *utility-based admission control*. The incoming traffic flow should not be accepted in the network if there are enough resources for the flow to get performance guarantees and if its acceptance will not deteriorate the performance of other, already accepted traffic flows beyond the agreed performance levels. It would be very efficient if the levels of the *performance deterioration* and the ability to provide the *requested service* would be uniformly assessed and defined. One of the ways this could be done is by using the connection utility.

An incoming flow could present its utility requirement to the network, which would then calculate by using some of the admission control procedures whether it is able to address that requirement.

Another idea is to simplify the admission control procedure in the following way. The network would calculate a *trunk reservation parameter*, which would be equal to the amount of bandwidth that can be accepted in the network so that the overall utility of the active traffic on the network would increase. Then the usual admission control procedure based on the available bandwidth would be performed, with the introduction of the trunk reservation parameter as additional control of the utility level. The algorithm for the calculation of the trunk reservation parameter exists, but is omitted in the Thesis, because the Thesis serves as just a basic definition of the utility-based dynamic bandwidth allocation scheme.

The use of utility as a performance measure presents a new approach, and there is yet a lot of space for interesting research to be done. Naturally, bandwidth is not the only network resource that should be taken under consideration when the level of satisfaction is being evaluated. Throughput is not the only parameter traffic connections are interested in. If we consider a multi-link network topology, this is even clearer. End-to-end delay, delay variation and packet loss rate are very important quality of service requirements. There is an interesting question of aggregating all of these requirements to calculate a single utility function, which would at least approximately contain all of these requirements. The work of Rajkumar et al

[LEE99][RAJ97][RAJ98] gives some insight at the level of that problem. Also, the work of Bouch et al [BOU00] analyses similar problems.

The research presented in this Thesis was focused on the problem of a single network link. A network link can never be observed as a completely independent system. A number of traffic flows that use our single network link traverse a number of other links and the bandwidth allocation for that flow is sometimes constrained by a bottleneck link somewhere in the network. That is why the first objective of the future research on Dynamic Bandwidth Partitioning must tackle the *end-to-end* implementation problem. It is even possible that new traffic classes will emerge from the traffic analysis in the multi-node heterogeneous network.

One of the possible ways in which this can be done is the introduction of *utility-based QoS routing*. QoS routing is routing the traffic in the network on the basis of the performance requirement of that traffic, rather then on the shortest-path basis. The main idea with utility-based routing is to use utility as a routing metric for a QoS-based routing mechanism. Each link should have the information about the utility a connection would acquire once it is accepted on that link. Based on that information, a routing algorithm would calculate the optimal route for the connection.

The routing decisions would be made in order to find an optimal route for the packet on the utility basis, and not to find a shortest path concerning the number of nodes, or the widest path concerning the bandwidth.

Internet routing protocols, such as Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) use the "shortest path routing" paradigm. Routers calculate shortest (i.e. least-cost) paths from source to destination. These paths are usually the paths with the smallest number of intermediate hops, but other routing metrics may be used. This is where utility could be used. Each link could be associated with the information of the utility level that would be generated for the traffic flow from a certain traffic class. Based on this information, the route calculation is done using shortest-path algorithms such as the Dijkstra algorithm.

It is interesting to analyse the use of utility as a routing metric further. In general, routing metrics [WAN96][RAK6] are the representation of a network in routing. They have major implication not only on the complexity of path computation, but also on the range of QoS requirements that can be supported. The metrics must reflect the basic characteristics of the network. The information they contain should make it possible to support basic QoS requirements. The metrics to some extent determine the types of QoS that the network can support, because all QoS requirements have to be mapped onto the constraints on a path expressed in terms of the metric. This analysis is similar to the analysis of the utility as a performance metric, and this shows that utility could be successfully used as a routing metric.

Utility is a *concave* routing metric, contrary to the *additive* routing metric such as delay, or *multiplicative* routing metrics such as packet loss probability. This means that the utility generated on a multi-link path through the network is determined by the minimal utility on the links that form the multi-link path.

7.3. Conclusion

There is an obvious need for improvement of the way IP-based Internet allocates it resources. IP does not provide any performance guarantees, and the end-users must rely on the appropriate traffic load patterns to receive the requested level of the network performance.

The objective of the research presented in this Thesis was to define and analyse a new bandwidth allocation scheme, which could be used in the integrated IP-based network to provide bandwidth allocation based on the QoS requirements. The objectives of the research have been fulfilled, and this Thesis provides a full definition of the scheme and the comprehensive analysis of the scheme implementation on the single link network model.

The main contribution of this work is in the introduction of the new scheme, which we call Dynamic Bandwidth Partitioning. This scheme uses dynamic control of bandwidth partitioning as the means to allocate the bandwidth. Further contribution is in the introduction of a new performance metric, connection utility. The scheme was designed to maximise the mean connection utility. Specially designed partitioning algorithm uses simple additive increase additive decrease linear control mechanism to update the bandwidth partitioning in order to increase the mean utility generated on the network.

Furthermore, the implementation of the scheme in different network environments, such as MPLS-enabled network and VPN was discussed in detail. A foundation for a new scheme, Hierarchical Dynamic Bandwidth Partitioning was defined in this Thesis, too.

There is still a variety of interesting research issues that will be a subject of future research. Some of the issues have been analysed in this Chapter. The most important work will consist of the research on the solutions for the end-to-end implementation of the DBP scheme. Furthermore, a lot of work needs yet to be done on the definition of utility functions, and of the possible ways to include other quality of service parameters in the utility function.

This research has been published in a number of publications. The list of the publications is given in the following section.

AUTHOR’S PUBLICATIONS

[RAK1] Veselin Rakocevic, John Griffiths, Graham Cope, “***Performance Analysis of Bandwidth Allocation Schemes in Multiservice IP Networks using Utility Functions”,*** International Teletraffic Congress, ITC17, Salvador da Bahia, Brazil, December 2001

[RAK2] Veselin Rakocevic, John Griffiths, Graham Cope, “***Dynamic Resource Management in Virtual Private Networks***”, FITCE Congress 2001, Barcelona, Spain, August 2001; also published in the Journal of the Institution of British Telecommunications Engineers, Vol 2, part 3, July-September 2001

[RAK3] Veselin Rakocevic, John Griffiths, Graham Cope, **“*Dynamic Partitioning of Link Bandwidth in IP/MPLS Networks***”, IEEE International Conference on Communications, ICC 2001, Helsinki, Finland, June 2001

[RAK4] Veselin Rakocevic, John Griffiths, Graham Cope, “***Linear Control in Dynamic Link Sharing for Multi-Class IP Networks***”, 17th IEE UK Teletraffic Symposium, Dublin, Ireland, May 2001

[RAK5] Veselin Rakocevic, John Griffiths, “***Value-based Evaluation of Strategies for Integration of Differentiated Internet Traffic Classes***”, 16th IEE UK Teletraffic Symposium, Harlow, UK, May 2000

[RAK6] Veselin Rakocevic, John Griffiths, “***Quality of Service Based Routing in TCP/IP Networks: The Missing Piece in Providing Service Guarantees***”, 2nd IEE Conference on Postgraduate Research in Electronics, Photonics and Related Fields, Nottingham, UK, April 2000

[RAK7] Veselin Rakocevic, John Griffiths, *“**Physical Separation of Bandwidth Resources for Differentiated TCP/IP Traffic***”, 8th International Conference on Telecommunication Systems, Nashville, USA, March 2000

REFERENCES

[ABD97] Tarek F. Abdelzaher, Ella M. Atkins, Kang G. Shin, "*QoS Negotiation in Real-Time Systems and its Application to Automated Flight Control*", IEEE Real-Time Technology and Applications Symposium, June 1997

[ALT97] Eitan Altman, Damien Artiges, Karim Traore *"On the Integration of Best-Effort and Guaranteed Performance Services"*, INRIA Research Report No.3222, July 1997

[APO98] G.Apostolopoulos, R.Guerin, S.Kamat and S.Tripathi, "*Quality of Service based Routing : A Performance Perspective*", Computer Communication Review, Volume 28, Number 4, October 1998

[ARV99] A.Arvidsson, P.Karlsson, "*On Traffic Models for TCP/IP"*, ITC16, Edinburgh, UK, June 1999

[ASH99] G.Ash, B.Jamoussi, Y.Lee, O.S.Aboul-Magd, "*QoS Resource Management in MPLS-based Networks*", Internet Draft, draft-ash-qos-routing-00.txt, February 1999

[ASH01] J. Ash, Y.Lee, P.Ashwood-Smith, B.Jamoussi, D.Fedyk, D.Skalecki, L.Li, "*LSP Modification Using CR-LDP*", Internet Draft, March 2001

[AWA00] Husam O. Awadalla, "*Resource Management for Multimedia Traffic over ATM Broadband Satellite Networks"*, PhD Thesis, QMW College, University of London, March 2000

[AWD01] Daniel O. Awduche, et. al., "*RSVP-TE: Extensions to RSVP for LSP Tunnels"*, Work in Progress, February 2001

[BAN00] Albert Banchs, Robert Denda, "*A Scalable Share Differentiation Architecture for Elastic and Real-Time Traffic"*, Eighth International Workshop on Quality of Service, IWQOS2000

[BLA95] U. Black, "*TCP/IP and Related Protocols",* McGraw-Hill, 1995

[BOD00] Eliane Bodanese, "*A Distributed Channel Allocation Scheme for Cellular Networks using Intelligent Agents",* PhD Thesis, QMW College, University of London, November 2000

[BOU00] A.Bouch, M.A.Sasse, H.DeMeer, "Of Packets and People: A User-centred Approach to Quality of Service", IWQOS2000, Monterey, CA, USA

[BRE00] Lee Breslau, Edward Knightly, Scott Shenker, Ion Stoica, Hui Zhang, "*Endpoint Admission Control: Architectural Issues and Performance*", in Proc. SIGCOMM 2000

[CAO99] Zhiruo Cao, Ellen W. Zegura*, "Utility Max-Min: An Application-Oriented Bandwidth Allocation Scheme*", INFOCOM '99

[CHE98] Shigang Chen, Klara Nahrstedt, "*An Overview of Quality of Service Routing for the Next Generation High-Speed Networks: Problems and Solutions*", IEEE Network, Special Issue on Transmission and Distribution of Digital Video, November/December 1998

[CHI89] Dah-Ming Chiu, Raj Jain, "*Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*", Computer Networks and ISDN Systems 17 (1989), 1-14

[COH] J.W.Cohen, "*The Multiple Phase Service Network with Generalized Processor Sharing*", Acta Informatica, Vol.12, pp. 245-284

[COST242] James Roberts, Ugo Mocci, Jorma Virtamo (Eds*.), "Broadband Network Teletraffic*", Final Report of Action COST 242, lecture Notes in Computer Sciences, Vol. 1155, Springer 1996

[CRO98] Jon Crowcroft, Phillipe Oechslin, *"Differentiated End-to-end Internet Services using a Weighted Proportional Fair Sharing TCP*", Computer Communication sReview, Vol. 28, 1998

[DUF99] N.G.Duffield, Pawan Goyal, Albert Greenberg, Partho Mishra, K.K.Ramakrishnan, Jacobus E. van der Merwe*, "A Flexible Model for Resource Management in Virtual Private Networks*", SIGCOMM99,

[FEI99] Sidnie Feit, "*TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security*", McGraw-Hill 1999

[FLO93] Sally Floyd, Van Jacobson, "*Random Early Detection Gateways for Congestion Avoidance*", IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August 1993, pp. 397-413

[FLO95] Sally Floyd, Van Jacobson*, "Link-Sharing and Resource Management Models for Packet Networks*", IEEE/ACM Trans. on Networking, Vol.3, No.4, August 1995

[FLO99] Sally Floyd, Kevin Fall*, "Promoting the Use of End-to-End Congestion Control in the Internet"*, IEEE/ACM Trans. on Networking, Vol.7, Iss. 4, August 1999

[FLO01] Sally Floyd*, "Report on Some Recent Developments in TCP Congestion Control*", January 2001, to appear in IEEE Comm. Magazine, April 2001

[GAR79] M.Garey, D.Johnson, "*Computers and Intractability, A Guide to the Theory of NP-Completeness*", W.H.Freeman and co., San Francisco, 1979

[GAR99] Rahul Garg, Huzur Saran*, "Scheduling Algorithms for Bounded Delay Service in Virtual Networks"*, in Proc. Globecom99

[GAR00] Rahul Garg, Huzur Saran*, "Fair bandwidth Sharing Among Virtual networks: A Capacity Resizing Approach"*, INFOCOM 2000, Tel Aviv, Israel, March 2000

[GIB00] R.J.Gibbens, S.K.Sargood, F.P.Kelly, H.Azmoodeh, R.Macfayden, N.Macfayden, "*An Approach to Service Level Agreements for IP Networks with Differentiated Services*", Article submitted to Royal Society, February 2000

[GIB] R. Gibbens, R. Mason, R. Steinberg, "*Multiproduct Competition Between Congestible Networks*", available at http://www.soton.ac.uk/ram2/papers.html

[GOY99] Pawan Goyal, Albert Greenberg, Charles R. Kalmanek, William T. Marshall, Partho Mishra, Doug Nortz, K.K. Ramakrishnan*, "Integration of Call Signalling and Resource Management for IP Telephony*", IEEE Network, Vol.13, no.3, pp 24-32, May/June 1999

[GRO85] Donald Gross, Carl M. Harris, "*Fundamentals of Queuing Theory*", John Wiley & Sons, 1985

[GUE99] R. Guerin and A. Orda. ``*QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms*.'', IEEE/ACM Transaction on Networking, Vol. 7, No. 3, June 1999, pp. 350-364.

[GUP95] Amit Gupta, Domenico Ferrari, "*Resource Partitioning for Real-Time Communication*", IEEE/ACM Trans. on Networking, Vol.3, No.5, October 1995

[HIG98] Gary Higginbottom, "*Performance Evaluation of Communication Networks"*, Artech House, 1998

[HOL00] Felicia Holness, "*Congestion Control Mechanisms within MPLS Networks",* PhD Thesis, QMW College, University of London, September 2000

[HUI98] Christian Huitema, "*IPv6, The New Internet Protocol*", Prentice Hall, 1998

[HUR99] Paul Hurley, Jean-Yves Le Boudec *"A Proposal for an Asymmetric Best-Effort Service*", 7th International Workshop on Quality of Service, London, June 1999

[IETF] Internet Enginnering Task Force, http://www.ietf.org

[JAC88] Van Jacobson, "*Congestion Avoidance and Control*", in Proc. ACM SIGCOMM, 1988

[JAM01] Bilel Jamoussi (ed.), "*Constraint-based LSP Setup using LDP"*, Work in Progress, February 2001

[JON95] Michael B. Jones, Paul J. Leach, Richard P. Draves, Joseph S. Barrera, III, "*Modular Real-Time Resource Management in the Rialto Operating System"*, Fifth Workshop on Hot Topics in Operating Systems, May 1995

[KEL97] Frank Kelly, "*Charging and Rate Control for Elastic Traffic"*, European Trans. on Telecommunications, Vol. 8, pp. 33-37, 1997

[KEL98] F.Kelly, A.Maulloo, D.Tan, "*Rate Control for Communication Networks; Shadow Prices, Proportional Fairness and Stability*", Journal of the Operational Research Society 49 , 1998

[KEY90] Peter Key, "*Optimal Control and Trunk Reservation in Loss Networks*", Probability in the Engineering and Informational Sciences, vol.4, 1990, pp. 203-242

[KEY99] P.B. Key, D.R. McAuley, "*Differential QoS and Pricing in Networks: Where Flow Control Meet Game Theory"*, IEE Proc.-Softw. Vol. 146, No.1, February 1999

[KIR99] P.Kirkby, et.al, "*The use of economic and control theory analogies in the design of policy based Dynamic Resource Control (DRC) network architectures*", ITC16, Edinburgh, UK, 1999

[KLE75] Leonard Kleinrock, "*Queuing Systems, Volume I: Theory",* John Wiley & Sons, 1975

[KLE76] Leonard Kleinrock, "*Queuing Systems, Volume II: Computer Applications"*, John Wiley & Sons, 1976

[LAW91] Averill M. Law, W. David Kelton, "*Simulation Modeling & Analysis"*, McGraw-Hill, 1991

[LEB] Jean-Yves Le Boudec, "*Rate Adaptation, Congestion Control and Fairness: A Tutorial"*

[LEE99] Chen Lee, John Lehoczky, Ragunathan (Raj) Rajkumar, Dan Siewiorek, "*On Quality of Service Optimization with Discrete QoS Options*", IEEE Real-Time Technology and Application Symposium, June 1999

[LIP77] S.Lippman, S.Stidham, "*Individual versus Social Optimization in Exponential Congestion Systems*", Operations Research, Vol.25, No.2, March-April 1977

[MAS98] L.Massoulie, J.Roberts, "*Bandwidth Sharing and Admission Control for Elastic Traffic*", ITC Specialist Seminar, Yokohama, 1998

[MAS99] L.Massoulie, J.Roberts, "*Bandwidth Sharing: Objectives and Algortihms*", IEEE INFOCOM 99, New York, USA

[McD00] David McDysan, "QoS & Traffic Management in IP&ATM Networks", McGraw-Hill, 2000

[McN00] Dylan McNamee, Charles Krasic, Kang Li, Ashvin Goel, Erik Walthinsen, David Steere and Jonathan Walpole, "*Control Challenges in Multi-Level Adaptive Video Streaming*", IEEE Conference on Decision anmd Control, CDC2000, Sydney, Australia, December 2000

[MIT96] Debasis Mitra, Ilze Ziedins, "*Virtual Partitioning by Dynamic Priorities: Fair and Efficient Resource-Shairng by Several Services*", Prco. 1996 International Zurich Seminar on Digital Communications, Lecture Notes in Computer Science, Broadband Communications, B.Plather (ed.), Springer, 1996, pp.173-185

[MIT97] Debasis Mitra, Ilze Ziedins*, "Hierarchical Virtual Partitioning: Algorithms for Virtual private Networking*", IEEE GLOBECOMM97, pp. 1784-1791

[MIT99] Debasis Mitra, John A. Morrison, K. G. Ramakrishnan, "*Virtual Private Networks: Joint Resource Allocation and Routing Design"*, in Proc. INFOCOM99

[MOY98] J. Moy, "*OSPF: Anatomy of an Internet Routing Protocol"*, Addison Wesley, 1998

[NAH98] Klara Nahrstedt, Shigang Chen*, "Coexistence of QoS and best-Effort Flows – Routing and Scheduling*", 10th IEEE Tyrrhenian International Workshop on Digital Communications: Multimedia Communications, Italy, September 1998

[NOR00] "*Quality of Service in Frame-Switched Networks*", Nortel White Paper, 2000

[NUN99] Rudesindo Nunez-Queija, Hans van den Berg, Michel Mandjes, *"Performance Evaluation of Strategies for Integration of Elastic and Stream Traffic*", ITC16, Edinburgh, UK, June 1999

[ODL99] Andrew Odlyzko, "*Paris Metro Pricing: The minimalist differentiated services solution*", International Workshop on QoS '99, London, June 1999

[ONV94] Raif Onvural, "*Asynchronous Transfer Mode Networks: Performance Issues*", Artech House, 1994

[PIT93] Jonathan M. Pitts, "*Cell-rate Simulation Modelling of Asynchronous Transfer Mode Telecommunications Networks",* PhD Thesis, QMW College, University of London, July 1993

[PIT00] Jonathan M. Pitts, John A. Schormans, "*Introduction to ATM and IP Design and Performance"*, J. Wiley & Sons, 2000

[PAR94] Colin Parris, Hui Zhang, Domenico Ferrari, "*A Dynamic Management Scheme for Real-Time Connections*", IEEE INFOCOM 1994

[PAR96] Kihong Park, Gitae Kim, Mark Crovella, "*On the Relationship between File Sizes, Transport Protocols and Self-similar Network Traffic",* International Conference on Network Protocols, 1996

[RAJ97] Ragunathan Rajkumar, Chen Lee, John Lehoczky, Dan Siewiorek, "*A Resource Allocation Model for QoS Management"*, IEEE Real-Time Systems Symposium, December 1997

[RAJ98] Ragunathan Rajkumar, Chen Lee, John P. Lehoczky, Daniel P. Siewiorek, "*Practical Solutions for QoS-based Resource Allocation Problems*", IEEE Real-Time Systems Symposium, 1998

[RFC793] J. Postel, "*Transmission Control Protocol"*, IETF Request for Comments, September 1981

[RFC1633] R.Braden, D.Clark, S.Shenker*, "Integrated Services in the Internet Architecture: an Overview*", IETF Request for Comments 1633, June 1994

[RFC2001] W. Stevens, "*TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*", IETF Request for Comments 2001, January 1997

[RFC2205] R.Braden, L.Zhang, S.Berson, S.Herzog, S.Jamin, "*Resource Reservation Protocol (RSVP) - Version 1 Functional Specification*", IETF Request for Comments 2205, September 1997

[RFC2211] J.Wroclawski, "*Specification of the Controlled-Load Network Element Service*", IETF Request for Comments 2211, September 1997

[RFC2212] S.Shenker, C.Partridge, R.Guerin, "*Specification of Guaranteed Quality of Service*", IETF Request for Comments 2212, September 1997

[RFC2386] E.Crawley, R.Nair, B.Rajagopalan, H.Sandick, *"A Framework for QoS-based Routing in the Internet*", IETF Request for Comments 2386, August 1998

[RFC2460] S.Deering, R.Hinden, "*Internet Protocol Version 6 (IPv6) Specification"*, IETF Request for Comments 2460, December 1998

[RFC2474] K. Nichols, S. Blake, F.Baker, D. Black, "*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers"*, IETF Request for Comments 2474, December 1998

[RFC2475] S. Blake, D.Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "*An Architecture for Differentiated Services",* IETF Request for Comments 2475, December 1998

[RFC2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "*Assured Forwarding PHB Group",* IETF Request for Comments 2597, June 1999

[RFC2598] V. Jacobson, K. Nichols, V. Poduri, "*An Expedited Forwarding PHB",* IETF Request for Comments 2598, June 1999

[RFC3031] E. Rosen, A. Viswanathan, R. Callon, "*Multiprotocol Label Switching Architecture*", IETF Request for Comments 3031, January 2001

[RFC3032] E.Rosen, D.Tappan, G.Fedorkow, Y.Rekhter, D.Farinacci, T.Li, A.Conta, "*MPLS Label Stack Encoding"*, IETF Request for Comments 3032, January 2001

[RFC3036] L.Amdersson, P.Doolan, N.Feldman, A.Fredette, B.Thomas, "*LDP Specification"*, IETF Request for Comments 3036, January 2001

[ROB99] L.Massouile, J.Roberts, "*Arguments in Favour of Admission Control for TCP Flows*", in Proc. ITC-16, Edinburgh, UK, June 1999

[ROS95] Keith W. Ross*, "Multiservice Loss Models for Broadband Telecommunication Networks*", Springer, 1995

[SAR99] Saswati Sarkar, Leandros Tassiulas*, "Fair Allocation of Utilities in Multirate Multicast Networks*", Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, Urbana, Illinois, USA, 1999.

[SHE95] Scott Shenker, "*Fundamental Design Issues for the Future Internet*", IEEE Journal on Selected Areas in Telecommunications, Vol.13, No.7, September 1995

[SHE98] Lee Breslau, Scott Shenker, *"Best-Effort versus Reservations: A Simple Comparative Analysis*", ACM Computer Communications Review, vol. 28, pp. 131-143, September 1998;

[STA98] William Stallings, "*High Speed Networks: TCP/IP and ATM Design Principles*", Prentice Hall, New Jersey, 1998

[STE94] W. Richard Stevens, "*TCP/IP Illustrated, Volume 1: The Protocols*", Addison-Wesley, 1994

[STE00] Peter Steenkiste, Qingming Ma "*Supporting Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm*", IEEE INFOCOM 99

[STE01] Robert Stewart, John Schormans, *"Accurate Dimensioning of Service Level Agreements in Voice over IP Networks*", 40th FITCE Congress, Barcelona, Spain, August 2001

[STO00] Ion Stoica, Hui Zhang, T.S. Eugene Ng, "*A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-time, and Priority Services*", IEEE/ACM Transactions on Networking, Vol.8, No.2, April 2000

[VAN99] B.Vandalore, W.Feng, R.Jain, S.Fahmy, "*A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia*", OSU Technical Report, OSU-CISRC-5/99-TR-14, available through www.cis.ohio-state.edu/¬jain/papers.html

[VDB00] Hans van den Berg, Michel Mandjes, "*Admission Control in Integrated Networks: Overview, Evaluation and Research Perspectives*", 8th International Conference on Telecommunication Systems, Nashville, USA, March 2000

[VER99] Dinesh Verma, "*Supporting Service Level Agreements on IP Networks*", Macmillan Technical Publishing, 1999

[WAN96] Z.Wang and J.Crowcroft, "*Quality of Service Routing for Supporting Multimedia Applications*", IEEE Journal Selected Areas in Communications, September 1996

[YAS83] S.F.Yashkov, "*A Derivation of Response Time Distribution for a M/G/1 Processor-Sharing Queue"*, Problems of Control and Inf. Theory, Vol.12 (2), 1983, pp.133-148

[ZHA95] Hui Zhang, "*Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks"*, proceedings of the IEEE, 83 (10), October 1995