RESOURCE DIMENSIONING IN A MIXED TRAFFIC ENVIRONMENT

By

Selwin Jakobus Emiel Roon

Submitted in the partial fulfillment of the requirements for the degree

Master of Engineering (Electronic Engineering)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORA

September 2004

Opsomming

'n Belangrike doelwit van moderne datanetwerke is om verskillende tipes toepassings oor 'n enkele netwerkinfrastruktuur te kan akkomodeer. Die kombinasie van data, spraak, video, en konferensie verkeer, met elk 'n unieke kwaliteitvereiste, maak die taak van bandwydte toekenning baie uitdagend. Om dienskwaliteit te probeer waarborg deur blote oor-voorsiening van bandwydte sal nie werk oor die lang termyn nie, aangesien netwerkhulpbronne duur is. Die mikpunt van deeglike bandwydtetoekenning is om die vereiste dienskwaliteit te bevredig en ter selfde tyd die toegekende bandwydte optimaal te benut. Die toekenningsparameters wat op die oomblik deur diensverskaffers gebruik word is gebaseer op aanbevelings wat nie noodwendig optimaal is nie.

Hierdie verhandeling fokus op die taak om effektiewe toekenningsparameters af te lei vir 'n gegewe netwerkargitektuur. Vier netwerkverkeerklasse is gedefinieer, naamlik Ware Tyd (WT), Interaktiewe Besigheid (IB), Massa Besigheid (MB) en Algemene Data (AD). Hierdie klasse moet onderskeidelik begroot word in terme van bandwydte en verkeersregulering. Om hierdie doel te bereik is die DiffServ meganisme en die kwaliteit vereistes van die klasse bestudeer. Die studie dui aan dat UDP die dominerende Laag 4 protokol vir die WT klas is, terwyl die res van die klasse meestal van TCP gebruik maak. Gevolglik is aparte analises uitgevoer om gepaste verkeersmodelle te identifiseer vir UDP en TCP.

Analise van ware netwerkdata toon dat moderne netwerkverkeer gekenmerk word deur lang-bereik afhanklikheid, self-soortgelykheid en 'n baie onreëlmatige aard. Evaluasies van verskeie netwerkverkeermodelle bewys dat die *Multi-fractal Wavelet Model* (MWM) die mees geskikte vir TCP is oor sy vermoë om lang-bereik afhanklikheid en self-soortgelykheid vas te vang. Die *Markov Modulated Poisson Process* (MMPP) is in staat om die gereelde lang AF-periodes en onreëlmatigheid teenwoordig in UDP verkeer, te modelleer. Hierdie twee modelle is dus in simulasies gebruik.

'n Geskikte simulasie model is geïmplementeer om die werkverrigting van die vier klasse te evalueer. Opgewekte verkeer is deur die simulasie model gestuur terwyl die vertraging en

verliese deurgaans gemeet is. Tydens enkel-klas simulasies is toekenningswaardes gemeet terwyl voldoende dienskwaliteit gehandhaaf is. Veelvoudige-klas simulasies het hierdie waardes geëvalueer onder die effek van statistiese multipleksering.

Simulasieresultate het numeriese waardes vir die toekenningsfaktore opgelewer. Hierdie toekenningsfaktore is gebruik om die betrokke netwerkskakel se data-tempo te bereken as 'n funksie van die gemiddelde vereiste bandwydte. Die gebruik van klas-gebaseerde differensiasie het getoon dat streng vertraging en verlieslimiete gewaarborg kan word, selfs wanneer die bandwydtebenutting baie hoog is (tot 90%). Simulasie resultate toon klein afwykings van beste-praktyk-aanbevelings: 'n waarde van 4 word huidiglik gebruik vir die WT en IB klasse, terwyl 'n waarde van 2 gebruik word vir MB. Hierdie verhandeling dui daarop dat 3.89 vir WT, 3.81 vir IB en 2.48 vir MB die gespesifiseerde dienskwaliteit meer akkuraat lewer. Daar is verder ook waargeneem dat die MB klas by tye onder presteer terwyl die WT en IB klasse oor presteer het. Die gevolgtrekking is gemaak dat twee moontlike oplossings vir die problem bestaan. Eerstens kan die bandwydte verspreiding tussen klasse hersien word; 'n groter persentasie moet aan die MB klas toegeken word. Tweedens kan die kwaliteit waarborge vir die MB klas gewysig word.

Die resultate dra by tot die proses van bandwydte toekenning deur waarde toe te voeg tot die toekenningsfaktore deur simulasies eerder as blote intuïsie of raaiskote.

Sleutelwoorde: Volgende Generasie Netwerke, Hulpbronbegroting, Dienskwaliteit, Vertraging, Verlies, Diensleweringsooreenkoms, Differensieërde Dienste, Diensklasse, Verkeersmodellering, Netwerksimulering.

Summary

An important goal of modern data networks is to support multiple applications over a single network infrastructure. The combination of data, voice, video and conference traffic, each requiring a unique Quality of Service (QoS), makes resource dimensioning a very challenging task. To guarantee QoS by mere over-provisioning of bandwidth is not viable in the long run, as network resources are expensive. The aim of proper resource dimensioning is to provide the required QoS while making optimal use of the allocated bandwidth. Dimensioning parameters used by service providers today are based on best practice recommendations, and are not necessarily optimal.

This dissertation focuses on resource dimensioning for the DiffServ network architecture. Four predefined traffic classes, i.e. Real Time (RT), Interactive Business (IB), Bulk Business (BB) and General Data (GD), needed to be dimensioned in terms of bandwidth allocation and traffic regulation. To perform this task, a study was made of the DiffServ mechanism and the QoS requirements of each class. Traffic generators were required for each class to perform simulations. Our investigations show that the dominating Transport Layer protocol for the RT class is UDP, while TCP is mostly used by the other classes. This led to a separate analysis and requirement for traffic models for UDP and TCP traffic.

Analysis of real-world data shows that modern network traffic is characterized by long-range dependency, self-similarity and a very bursty nature. Our evaluation of various traffic models indicates that the Multi-fractal Wavelet Model (MWM) is best for TCP due to its ability to capture long-range dependency and self-similarity. The Markov Modulated Poisson Process (MMPP) is able to model occasional long OFF-periods and burstiness present in UDP traffic. Hence, these two models were used in simulations.

A test bed was implemented to evaluate performance of the four traffic classes defined in DiffServ. Traffic was sent through the test bed, while delay and loss was measured. For single class simulations, dimensioning values were obtained while conforming to the QoS

specifications. Multi-class simulations investigated the effects of statistical multiplexing on the obtained values.

Simulation results for various numerical *provisioning factors* (PF) were obtained. These factors are used to determine the link data rate as a function of the required average bandwidth and QoS. The use of class-based differentiation for QoS showed that strict delay and loss bounds can be guaranteed, even in the presence of very high (up to 90%) bandwidth utilization. Simulation results showed small deviations from best practice recommendation PF values: A value of 4 is currently used for both RT and IB classes, while 2 is used for the BB class. This dissertation indicates that 3.89 for RT, 3.81 for IB and 2.48 for BB achieve the prescribed QoS more accurately. It was concluded that either the bandwidth distribution among classes, or quality guarantees for the BB class should be adjusted since the RT and IB classes over-performed while BB under-performed.

The results contribute to the process of resource dimensioning by adding value to dimensioning parameters through simulation rather than mere intuition or educated guessing.

Keywords: Next Generation Networks, Resource dimensioning, Quality of Service, Delay, Loss, Service Level Agreement, Differentiated Services, Classes of Service, Traffic Modelling, Network Simulation

Table of contents

PAGE

Abbreviations

ACF	-	Autocorrelation Function
AF	-	Assured Forwarding
BB	-	Bulk Business (data)
BE	-	Best Effort
CE	-	Customer Edge Router
CSR	-	Combined Server Rate
DiffServ	-	Differentiated Services
DSCP	-	DiffServ Code Point
EF	-	Expedited Forwarding
GD	-	General Data
GS	-	Guaranteed Services
IB	-	Interactive Business (data)
LRD	-	Long Range Dependency
LSR	-	Label Switched Router
MMPP	-	Markov Modulated Poisson Process
MPB	-	Maximum Permitted Burst
MWM	-	Multi-fractal Wavelet Model
NGN	-	Next Generation Network
PE	-	Provider Edge Router
PF	-	Provisioning Factor
PHB	-	Per Hop Behaviour
POTS	-	Plain Old Telephone System
QoS	-	Quality of Service
RT	-	Real Time (data)
SRD	-	Short Range Dependency
ToS	-	Type of Service

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

The *Next Generation Network* (NGN) is on our doorstep. We are entering a new era of communications and will witness the transformation from Internet Protocol version 4 (IPv4) to IPv6 in the near future. This transformation will resolve the worldwide addressing problem, improve scalability and allow a greater diversity of protocols to be used over a single network. The increased diversity of protocols is specifically aimed at supporting multiple applications over a single network infrastructure. Voice, video, data, conferencing, among others will be carried over a single network, using the same packet-based transmission.

The diversity of applications running over a single network has different measures of merit. Real-time data, such as voice traffic, does not need a very high throughput to deliver adequate performance. It can even tolerate some packet loss, but the delay bound is very strict. Other applications, for example the transmission and update of large databases, require high throughput and strict loss bounds, but the latency does not have a significant negative effect. The combination of these various performance specifications of a service are known as the *Quality of Service* (QoS).

QoS is provided for various applications by means of QoS mechanisms. These mechanisms are implemented on the network infrastructure to classify application traffic and to provide distinctive handling of the classes across the network. Over the last decade various QoS mechanisms were proposed, but until recently there was no real need for it. It is only since the

introduction of multimedia and real-time traffic to packet networks, that it became necessary to implement QoS mechanisms.

Services that are being offered on the described integrated network are referred to as broadband services. The provision of these services became feasible with the availability of larger bandwidth, hence the name, broadband services. The research presented in this dissertation was motivated by the optimization process of a particular broadband service: *Virtual Private Networks* (VPNs). Frame based VPNs have existed for more than 20 years [1].

Throughout the years, it was implemented in different ways. Most of the literature on VPNs is based on the security aspects of remote access. This study investigates the guarantees, utilization figures, and dimensioning to improve costs and profitability. Broadly speaking, VPNs can be categorized into three classes, namely leased line, layer 2 based, and layer 3 based VPNs.

Leased line VPNs are very expensive and do not scale well due to the dedication of a specific part of network infrastructure to a single customer. Consequently it is not very popular. Legacy layer 2 VPNs provided sufficient quality guarantees due to their connection-orientated nature. The drawback of this type of VPN is the requirement of its permanent virtual circuit (PVC). This becomes costly and does not scale well. For a bigger market share, a scalable and cheaper VPN solution is sought. This is where the Internet, with its global accessibility and cost effectiveness comes into play.

Layer 3 VPNs overcome the cost and scalability limitations of leased line and layer 2 VPNs. However, QoS guarantees are only achieved by static over-provisioning, which is not very popular today. Statistical multiplexing allows over-provisioning, but the choice of dimensioning values is crucial, since it directly affects the QoS, the cost, and the profit margin for the service provider.

Network provisioning is generally based on simple rules of thumb while considerable effort is being devoted to the development of a variety of QoS mechanisms. In this dissertation the

current mechanisms are analyzed, evaluated, and an effort is made to improve the values that are used for resource allocation.

This dissertation is focused on *network traffic theory*. In this context the concept of traffic theory refers to the application of mathematical and statistical techniques to evaluate the *traffic- performance relation*. This relation links traffic profiles, network capacity, resource utilization, and the achieved performance.

Although the research was motivated by VPNs, the results can be applied to any broadband services offered on integrated networks requiring service differentiation. This dissertation presents a comprehensive picture by highlighting and investigating various QoS enabling technologies, protocols, and network traffic characteristics from recent research and engineering work. Simulations are performed to verify theoretical models and results.

1.2 PROBLEM STATEMENT

The addition of real-time applications to packet switched networks implies that the nature of the service offered by the network can no longer be best effort. Bandwidth needs to be reserved for some applications. Methods for bandwidth reservation exist, but the question is, how much bandwidth and other resources are needed for different kinds of applications. Over-provisioning result in low resource utilization, which has inherent cost implications. On the other hand under-provisioning degrades the performance of real-time applications that are running over the concerned network.

No rigid mathematical methods can be used for dimensioning purposes due to the chaotic nature of the network traffic. Furthermore, QoS in packet switched networks is a relatively new concept, and literature on dimensioning is scarce. Consequently, vendors use experience and best practice recommendations to provision their networks. These dimensioning values are not necessarily optimal.

Taking the above into account, this dissertation focuses on the problem of intuitive guessing that is used for dimensioning in QoS enabled networks.

1.3 OBJECTIVE OF THE STUDY

The objective of this study is to derive and investigate suitable dimensioning values for various traffic classes, by using appropriate QoS specifications and suitable traffic models representative of real traffic. This goal will be achieved by analyzing the dynamic behaviour of various QoS parameters by means of simulations. This will include the following:

- Investigate the concept of QoS, how it can be measured, and how to relate specific QoS parameters to given applications.
- Investigate mechanisms to obtain QoS as well as motivating a specific mechanism to be used in simulations.
- Study the network architecture and protocols involved in various QoS mechanisms.
- Investigate the characteristics of network traffic in terms of statistical and mathematical descriptions.
- Evaluate relevant traffic models and determine the most effective models for traffic simulation purposes.
- Develop a simulation test bed involving the following:
    - The use of appropriate traffic models to simulate user demands for network resources.
    - Simulation of the relevant node architecture.
    - Measurement and evaluation of output data generated by simulation.
    - Derivation of suitable dimensioning values based on QoS parameter budgets.
- Compare results with other values used in practice, and
- Evaluate the results.

1.4 CONTRIBUTION

The main contribution of this dissertation is to provide sufficient QoS to mixed traffic in distinctive classes, while keeping the utilization as high as possible. This is achieved by using an appropriate functional architecture and a resource dimensioning method. Both the architecture and dimensioning method are relatively new and therefore leave room for improvement. Dimensioning values are currently based on intuition, lab tests and best practice recommendations. This dissertation contributes towards obtaining the most sufficient dimensioning values through simulations, rather than by the above mentioned methods. These simulations account for real traffic characteristics, QoS requirements, class based differentiation, and maximizing resource utilization.

1.5 DISSERTATION OUTLINE

In Chapter 2 various aspects regarding the provision of QoS in present day networks are investigated. The concepts of QoS are explained and levels as well as the parameters of QoS are described. QoS mechanisms, some limitations, and tradeoffs between the various mechanisms are also discussed. The choice of a certain mechanism to be used in simulation is motivated.

A detailed description of the network architecture in question is given in Chapter 3. The *Open System Interconnection* (OSI) reference model is reviewed, as some explanations and reasoning will refer to this model. A short summary of relevant protocols and aspects of packet overhead is provided. Details of the domain, node architecture and functionality, of a particular QoS mechanism; i.e. *Differentiated Services* (DiffServ), is discussed to serve as a foundation for simulations.

Chapter 4 provides an analysis of traffic characteristics. Models to synthesize network traffic are evaluated and the most appropriate ones selected for use in simulations.

Chapter 5 describes the experimental setup together with the considerations needed to perform adequate simulations. Simulation results are presented in chapter 6. Chapter 7 concludes this dissertation.

CHAPTER 2
QUALITY OF SERVICE

2.1 INTRODUCTION

Previously, voice traffic was carried on circuit switched networks, which, by their nature, guarantee QoS in terms of delay and throughput. Data, on the other hand was carried on separate packet switched networks, such as the Internet, that are best-effort services. However, in recent years there has been a convergence of heterogeneous traffic onto a single network. This integrated network is referred to by many as the *Next Generation Network* (NGN) [2], [3], [4].

Convergence is driven by the following two factors. Firstly, there is convergence with an aim to bring together different types of traffic onto a single link, rather than providing different links for each traffic type. Secondly, there is convergence at the application level of video conferencing, chat rooms and Multimedia Messaging Service (MMS), to name but a few. In either case, users expect the same QoS as if all the traffic had been carried on the traditional separate networks.

2.2 THE SERVICE LEVEL AGREEMENT

The described multimedia services must first be sold by any vendor providing such services. As can be expected, customers would want to know what they are buying and how much they are paying for it. For this reason the vendor and customer agree upon a service which is then documented, known as the *Service Level Agreement* (SLA).

The SLA is a contract between the provider and the user, which specifies the level of service that is to be provided during its term. SLAs are used by vendors and customers as well as internally by Information Technology (IT) shops and their end users. They can specify bandwidth availability, response times for routine and ad-hoc queries and response time for problem resolutions, including when the network is down and machine failure. SLAs can be very vague or extremely detailed, including the steps to be taken in the event of network failure.

Regarding legacy network services and applications, most people know exactly what is available and what they will pay for it. The diversity of modern multimedia applications with different guarantees makes the offering and pricing of these services a more challenging task. Furthermore, vendors try to cater for individual customer requirements, which add more complexity to service packages. For these reasons, the vendor listens to the customer's needs and then formulates a legal document known as the SLA to clearly state which services will be provided and consequently at what cost.

2.3 THE CONCEPT OF QOS

The concept of QoS is often mistaken for Availability of Service (AoS). AoS is the time periods during which a specific service is available. QoS, on the other hand, is the quality of the service during the time in which it is available [5].

In general, QoS is understood as the quantitative description of network characteristics including transmission rates, error rates, delay and other. These parameters can be measured, improved, and to some extent, guaranteed in advance [6]. QoS is not a characteristic of one layer of the protocol stack, but instead, is delivered over several layers and can be defined on different levels [5]:

- **Packet level:** Most of the quantitative QoS parameters are found on this level. Amongst others, parameters like delay, jitter, throughput and error rates are included. These parameters are mainly affected by network resources, such as available bandwidth, buffer space and the parameters of the concerned access protocol.

- **Circuit level**: During network congestion no new virtual circuits are created, known as call blocking, in order to re-establish proper functioning of the network. Call blocking for new, as well as existing calls, is included in this level. The two most important circuit level attributes are call routing and local management.
- **Transaction level**: The performance of one completed transaction is considered at this level. Packet loss rate and the ordered delivery of all the packets belonging to a specific transaction have a direct effect on the transaction time, which is considered the most important QoS parameter of the transaction level.
- **User level:** QoS on this level depends on the combination of user mobility and application type. Even with adaptive applications, often new locations may not support the minimum QoS.

Considering the objective of this study mentioned in Chapter 1, the focus of this dissertation is on the packet level of QoS, since this is the level that most directly relates to the physical dimensioning of the network resources. The circuit level performance is mainly affected by the degree to which customers fulfill their SLAs. The transaction level evaluates complete transactions, which are important only to a very small group of traffic sources served across modern day integrated networks. User level QoS is affected by application layer protocols, as well as the support of these protocols at different mobile locations. To deal with QoS, network links are categorized according to the following attributes:

- **Traffic characteristics**, specified in terms of bandwidth namely peak rate, minimum acceptable rate, average rate and maximum burst size.
- **Reliability requirements** of a session, for example, bit-error-rate, frame-error-rate and maximum packets loss ratio.
- **Delay requirements** such as maximum tolerable delay and maximum delay variation (jitter).

These attributes correspond to four basic performance parameters: delay, jitter, throughput and loss. These basic performance parameters can be analyzed separately and improved according to their relative importance as an engineering optimization problem. These parameters, along with their significance in various applications will be discussed in the following section.

2.4 QOS PARAMETERS

2.4.1 Delay

Delay, or *latency*, is the amount of time that it takes for a packet to be transmitted from one point in a network to another. These two points can vary, depending on the relevance, for example, of the time a packet traverses between an origin-destination pair. Another commonly used time measurement is the *round-trip-time* (RTT). The most significant forms of delay include queuing, serialization, propagation and forwarding delay [7].

Queuing Delay

Queuing delay is the amount of time that a packet has to wait in a queue while other packets ahead of it are being transmitted. It also involves the delay encountered as a result of statistical multiplexing. The queuing delay at a given router can vary over time: between zero seconds for an un-congested link, to the total time it takes to transmit all packets that are queued ahead of it. This form of delay contributes a significant amount to the total delay. Large buffers can result in longer queues, increasing the waiting time for a packet to get transmitted. During periods of congestion, queue memory management and scheduling disciplines allow one to control queuing delay on a per-class basis. However, queue memory management implies call blocking, which is undesirable. Such congestion avoidance algorithms will not be included in simulation since dimensioning is particularly aimed to provide sufficient resources.

Serialization Delay

Serialization delay is the amount of time that is required to place the bits of a packet onto the wire when a router transmits a packet. Serialization delay is measured in milliseconds. It is directly related to the size of the packet and the speed of the connection. Packet size can only be controlled by manipulation of the *maximum transmission unit* (MTU) or by forcing packet fragmentation. This results in higher overhead. The only effective way to reduce serialization delay is to install higher-speed router interfaces.

In a network consisting of high-speed interfaces, serialization delay contributes an insignificant amount to the overall end-to-end delay. However, in a network consisting of low-speed interfaces, serialization delay can contribute significantly to the overall end-to-end delay. Consequently, services with strict delay requirements cannot be sold in small portions. Typically a minimum of 16 kbps, and increments thereof, are sold. Otherwise the serialization delay becomes too large relative to the queuing delay. In chapter 5, serialization delay is tabulated in terms of the line speed and frame size. These figures are used to argue why minimum amounts of bandwidth have to be allocated to certain services.

Propagation Delay

Propagation delay is the amount of time that it takes for electrons or photons to traverse a physical link. The propagation delay is based on the speed of light. When estimating the propagation delay across a point-to-point link, one can assume one millisecond of propagation delay per 160 km round-trip distance. Propagation delay can not be changed because one has little control over the speed of light in optical fiber. It is interesting to note that the speed of light in optical fiber is approximately 65 percent of the speed of light in a vacuum, while the speed of electron propagation through copper is slightly faster, at 75 percent of the speed of light. Although the signal representing each bit travels slightly faster through copper than fiber, fiber has numerous advantages over copper. It results in fewer bit errors, supports longer cable runs between repeaters, and allows more bits to be packed into a given length of cable. Propagation delay forms a significant part of the total delay over long distances. The distance between the end users plays an important role for delay-sensitive applications, such as voice. A typical accepted maximum mouth to ear delay is 150ms [8], [9]. A call from South Africa to the UK would have a propagation delay of 34.4ms, which is almost a quarter of the total delay budget. On the other hand, a call from Pretoria to Johannesburg shows a propagation delay of 0.2ms, which is insignificant compared to the total delay budget.

Forwarding Delay

Forwarding delay is the amount of time that it takes a router to receive a packet, make a forwarding decision, and then begin transmitting the packet through an un-congested output

port. This represents the minimum amount of time that it takes the router to perform its basic function and is typically measured in tens or hundreds of microseconds. Other than deploying the industry standard in hardware-based routers, one has no real control over forwarding delay. This form of delay is very small and has no significant negative effect on any services offered in today's data networks.

These delay factors are a direct result of the architecture, available resources and physical medium along the network path. Other less significant factors, which are not discussed in this dissertation, include the following:

- The performance bottlenecks within hosts and servers,
- Operating system scheduling delays,
- Application resource contention delays,
- Physical layer framing delays,
- CODECs, compression, and packetization delays,
- The quality of the different TCP/IP implementations running on the end systems, and
- The stability of routing in the network.

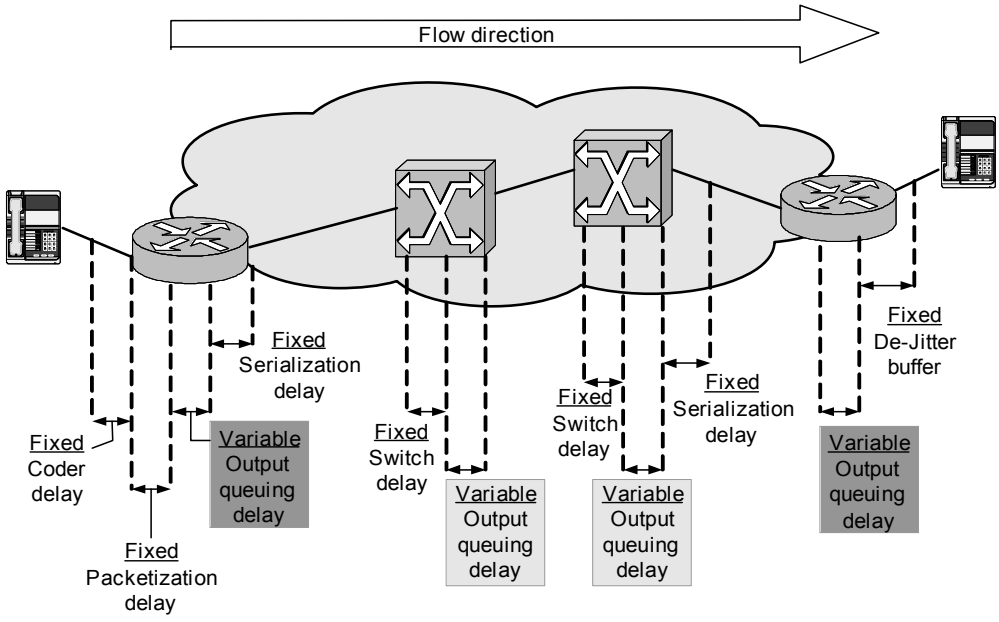


Figure 2.1 Delay experienced by a packet over the total end-to-end path

Only some factors of delay can be controlled by dimensioning. Others are inherent to equipment and physics. Figure 2.1 depicts a simplified illustration of the most significant forms of delay contributing to the total delay encountered by a packet traversing a network.

As indicated in figure 2.1 the only variable form of delay is the queuing delay. This is the case for a fixed server rate. Serialization delay can be varied, but only by changing the server rate. However, queuing delay can be improved with dynamic bandwidth sharing.

Therefore, when delay is analyzed later in this dissertation, only these forms of delay are taken into consideration, as they are the only factors that resemble significance and controllability in terms of resource dimensioning.

Managing Delay While Maximizing Bandwidth Utilization

Except for serialization delay, directly related to the server rate, queuing delay is the only components of end-to-end delay which is controllable. Support for differentiated service classes is based on managing the queuing delay experienced by different traffic classes during periods of network congestion. In the absence of active queue management techniques, such as Random Early Detection (RED) [10], there is a direct relationship between the bandwidth utilization on a link and the RTT delay. If a 5-minute weighted bandwidth utilization of 10 percent is maintained, there will be minimal packet loss and minimal RTT delay, because the output ports are generally underutilized. However, if the 5-minute weighted bandwidth utilization increases to approximately 50 percent, the average RTT starts to increase exponentially as shown in figure 2.2.

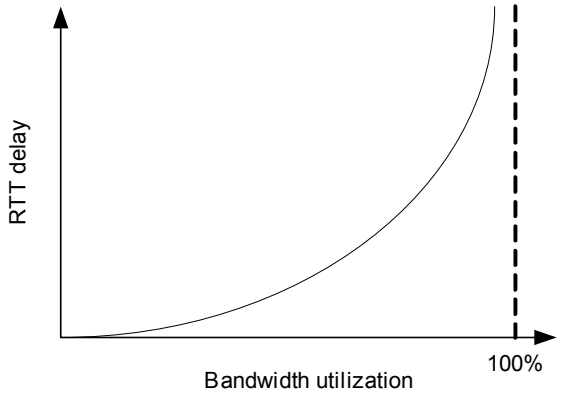


Figure 2.2 Bandwidth Utilization vs. Round-trip Time (RTT) Delay

When trying to manage delay, bandwidth utilization also needs to be maximized simultaneously for financial reasons. In a well-designed and properly dimensioned network, queuing delay should approach zero, when measured over time. There will always be extremely short periods of congestion, but network links need to be properly provisioned. Otherwise, queuing delay will increase rapidly, as the under-provisioned link is required to carry too much traffic.

2.4.2 Jitter

Jitter is the variation in delay over time, experienced by consecutive packets that are part of the same flow, as shown in figure 2.3. Jitter can be measured by a number of different techniques, including the mean, standard deviation, maximum, or minimum of the inter-packet arrival times for consecutive packets in a given *flow*. A network flow is defined as a unidirectional sequence of packets between given source and destination endpoints.

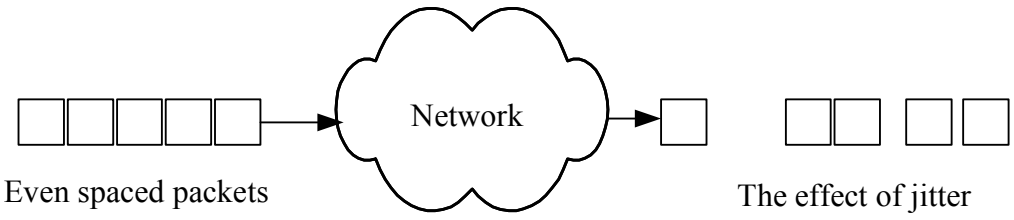


Figure 2.3 Jitter Makes Packet Spacing Uneven

Time Division Multiplexing (TDM) systems can cause jitter, but the variation in delay is so small that it can be ignored for all practical purposes. In a statistically multiplexed network the primary source of jitter is the variability of queuing delay over time for consecutive packets in a given flow. Another potential source of jitter is that consecutive packets in a flow may not follow the same physical path across the network due to equal-cost load balancing or routing changes.

Jitter increases exponentially with bandwidth utilization, similar to delay. There are a number of other considerations relevant to jitter in statistical multiplexing networks, where the end-to-end jitter is never constant. This is because the level of congestion in a network is constantly changing. Unless one is assured that the transmission of a packet will begin immediately after a router's forwarding decision, the amount of delay introduced at each hop in an end-to-end path is variable.

Impact of Jitter on perceived QoS depends on the application. Some applications are unable to handle jitter:

- With interactive voice or video applications, jitter can result in a jerky or uneven quality of the sound or image. The solution is to provide sufficient resources. Proper dimensioning includes adequate bandwidth allocation, appropriate traffic conditioning and a suitable scheduling discipline. The jitter which remains can be handled by a short playback buffer on the destination host that buffers packets briefly before replaying them as a smoothed data stream.
- For emulated TDM service over a statistical multiplexed network, jitter outside of a narrowly defined range can introduce errors. The solution is to perform proper dimensioning. Appropriate queuing disciplines and traffic condition are needed at the edges of the network keep jitter within a predefined range.

However, jitter affects only a small number of applications. For most applications, such as those running over the *Transport Control Protocol over Internet Protocol* (TCP/IP), jitter is not a problem. Also, for non-interactive applications such as streaming voice or video, jitter

does not present serious problems, because it can be overcome by using large playback buffers.

2.4.3 Throughput

Throughput is a generic term used to describe the capacity of a system to transfer data. It is easy to measure the throughput for a TDM service, because the throughput is simply equal to the bandwidth of the transmission channel. For example, the throughput of a DS-3 circuit is 45 Mbps. However, for TCP/IP statistically multiplexed services, throughput is much harder to define and to measure, because there are numerous ways that it can be calculated, including the byte or packet rate:

- Across the circuit,
- Of a specific application flow,
- Of host-to-host aggregated flows, or
- Of network-to-network aggregated flows.

The most direct way that a router's statistical multiplexing can be adapted to optimize throughput, is by varying the bandwidth that is allocated to different types of packets. In classic best-effort service, the router does not specifically control the bandwidth assigned to different traffic classes. Instead, during periods of congestion, all packets are placed into a single *first-in, first-out* (FIFO) queue. When faced with congestion, *User Datagram Protocol* (UDP) flows continue to transmit at the same rate, but *Transport Control Protocol* (TCP) flows detect and then react to packet loss by reducing their transmission rate. As a result, UDP ends up consuming the majority of the bandwidth on the congested port, but each TCP flow receives roughly an equal share of the remaining bandwidth.

When attempting to support differentiated treatment for a number of traffic classes, each class of traffic can be given different shares of output-port bandwidth. For example, a router can be configured to allocate specific amounts of bandwidth among traffic classes on the output port. One class of traffic can be given strict priority, but then a bandwidth limit is needed to prevent the starvation of the other classes. The support of differentiated service classes implies the use

of more than just a single FIFO queue on each output port. Various mechanisms were specifically designed for this purpose and will be discussed later in this chapter.

2.4.4 Loss

Loss occurs when a transmitted packet never reaches its destination. Three sources of packet loss in packet switched networks are defined:

- A break in a physical link that prevents the transmission of a packet,
- A packet that is corrupted by noise and is detected by a checksum failure at the downstream node, and
- Network congestion that leads to buffer overflow.

Firstly, breaks in physical links do occur, but they are rare, and the combination of self-healing physical layers and redundant topologies respond dynamically to this source of packet loss. Secondly, the chance of packet corruption is statistically insignificant, hence this source of packet loss are often ignored. This can be stated with the exception of wireless networking, only when using modern physical layer technologies.

Consequently, the primary reason for packet loss in a non-wireless IP network is due to buffer overflow, resulting from congestion. The amount of packet loss in a network is typically expressed in terms of the probability that a given packet will be discarded by the network.

IP networks do not carry a constant load, as traffic is bursty[1], which causes the load on the network to vary over time. There are periods when the volume of traffic that the network needs to carry exceeds the capacity of some of the components in the network. When this occurs, congested network nodes attempt to reduce their load by discarding packets. When the TCP/IP stack on host systems detects a packet loss, it assumes that the packet loss is due to congestion somewhere in the network.

[1] Bursty – A term to describe the nature of network traffic. A burst in network terms is known as a very high traffic load for a short period of time. Bursts occur at different time scales.

Packet loss in an IP network is not necessarily a bad thing. Each TCP session requires as much bandwidth as possible for its flow, without causing sustained congestion in the network. TCP accomplishes this by slowly transmitting at the beginning of each session, and then increasing the transmission rate until it eventually detects the loss of a packet. Since TCP associates packet loss with congestion at some point in the network, it reacts by temporarily reducing the transmission rate of the flow. Given enough time, each TCP flow will eventually settle on the maximum bandwidth it can get across the network without experiencing sustained congestion. When multiple TCP flows do this simultaneously, the result is fairness for all TCP sessions across the network. Thus, occasional packet loss is necessary as each TCP session needs to experience some packet loss to establish the maximum available bandwidth for a concerned flow.

2.5 THE DEVELOPMENT OF QOS MECHANISMS

Since the need for a QoS mechanism only recently became important, research and development in this field is still immature, and no single adequate mechanism has been developed yet. The various proposals all show strong and weak characteristics.

2.5.1 ToS Byte in the IP Header

The very first approach in 1981 was the definition of the IP Type of Service (ToS) byte in the IP header. Figure 2.4 shows the IP ToS byte with its fields.

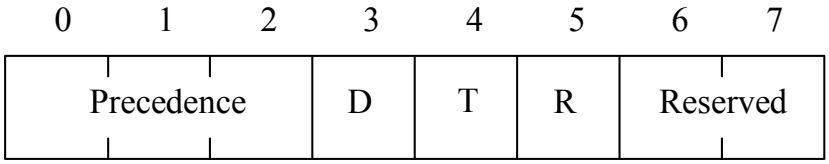


Figure 2.4 The ToS Byte in the IP Header

The precedence bits can be set by a node to select the relative priority of the packet. The three bits following the precedence bits are set to specify normal or low delay (D), normal or high throughput (T), and normal or high reliability (R). The final two bits of the ToS byte are reserved for future use.

Very little infrastructure was provided to support the delivery of differentiated service classes using this capability. Until the mid 1990s, the only practical application of this approach was the selective packet discard (SPD) feature.

2.5.2 IntServ

Comprehensive work by the IETF began in 1993, to provide more than a best effort service: Integrated Services (IntServ). The objective was to provide support for real-time data and non-real-time data simultaneously on a shared network. IntServ failed as a standalone QoS mechanism due to the fact that it was not scalable. All end systems have to support the Resource Reservation Protocol (RSVP), and all nodes in the network path have to support the IntServ model. A host uses RSVP to request a specific QoS from the network, on behalf of an application data stream. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream.

2.5.3 Enhancement to IntServ

An enhancement to the IntServ was defined in September 2001. It defined procedures that allow a single RSVP reservation to aggregate other RSVP reservations across a large IP network. It enhances the scalability of RSVP for use in large IP networks by reducing the number of signalling messages exchanged and the number of reservation states. Further, it streamlines the packet classification process in core routers and simplifies the packet queuing and scheduling by combining the aggregate streams into the same queue on an output port.

2.5.4 DiffServ

In the mid 1990s, alternative approaches were being considered to support more than a best effort service. The aim was to develop a scalable mechanism. In March 1998, the Internet Engineering Task Force (IETF) created the DiffServ Working Group after realizing that IntServ was not going to be deployed in production networks. The complete DiffServ architecture is defined in RFC 2475 [11]. DiffServ can be implemented on both IPv4 and IPv6.

In IPv4 the ToS octet [12] is used to host the DiffServ Code Point (DSCP). In IPv6 the Traffic Class octet [13] is used for the DSCP. The DSCP of each packet is set according to the relative importance of the packet's source or its application level protocol. Throughout the DiffServ domain, the packet's Per-Hop-Behaviour (PHB) is determined by the DSCP. DiffServ is not capable of providing the same strict QoS guarantees as IntServ; however the biggest problem with IntServ is scalability, which is solved with DiffServ.

2.5.5 Traffic Engineering

The need for Traffic Engineering becomes clear with the understanding of the other QoS approaches. These approaches all show some limitations and drawbacks. Network engineering can be seen as the process of placing the bandwidth where the traffic is, while Traffic Engineering is the process of placing the traffic where the bandwidth is. Traffic Engineering applications have an end-to-end view of network traffic flow, and the mapping of flows to resources. This holistic view enables flows to be intelligently routed through the network, ensuring that bandwidth is used efficiently and that QoS can be guaranteed throughout the network.

2.6 CHAPTER SUMMARY

QoS is considered as the performance, or quality, of a number of different network traffic measurements. The type of service determines which of the QoS parameters, and to what degree they, apply to the concerned service. QoS is achieved by implementing a QoS mechanism in a network.

Various QoS mechanisms have been discussed in this chapter. No single QoS mechanism to guarantee adequate performance along with efficient bandwidth utilization has been developed yet. This leaves room for research and development. An adequate mechanism will be a combination of the advantages of the various mechanisms. In this optimization process, some other factors contributing to QoS must also have to be considered. These factors form the building blocks to support differentiated services in IP networks and are mainly aimed at minimizing congestion, as congestion is the main culprit degrading QoS.

Various proposals regarding the ultimate QoS mechanism are being developed. At this stage, it seems that the ultimate mechanism will result as the "marriage" of routing and the appropriate QoS mechanism. This research does not propose another QoS mechanism, but rather focuses on DiffServ, which is currently being deployed the most widely, and seems to be most promising.

A more detailed description of DiffServ will be provided in chapter 3 in which the architecture is analyzed with the aim of drawing up mathematical models. These models are used to build a test bed to perform simulations to improve resource utilization.

CHAPTER 3
ARCHITECTURE AND DIMENSIONING

3.1 INTRODUCTION

Numerous network architectures and topologies exist in practice today, and it is very difficult to do an analysis that applies to all of them. As described in Chapter 2, DiffServ is currently the prominent QoS mechanism. In this chapter, DiffServ as a QoS mechanism is investigated to form a foundation of the functionality to perform dimensioning tasks. The following key aspects will be covered:

- The layered structure of the OSI reference model and the effect on dimensioning,
- overhead considerations with regards to dimensioning,
- DiffServ's dependence on IP and MPLS
- The DiffServ Domain,
- The DiffServ node architecture (service router functions),
- Per Hop Behaviour (PHB) and forwarding, and
- Resource dimensioning.

3.2 THE OSI MODEL, LAYERS AND THEIR RELEVANCE

The *Open Systems Interconnection* (OSI) reference model and the TCP/IP protocol stack are well known. This section investigates on which layers of the protocol stack the functions are being performed. Figure 3.1 illustrates a point to point connection, making use of only one router.

Under normal operation, routers do not perform tasks of any layer higher than the Network layer. Functions and operation of higher layers that affect dimensioning tasks are briefly described.

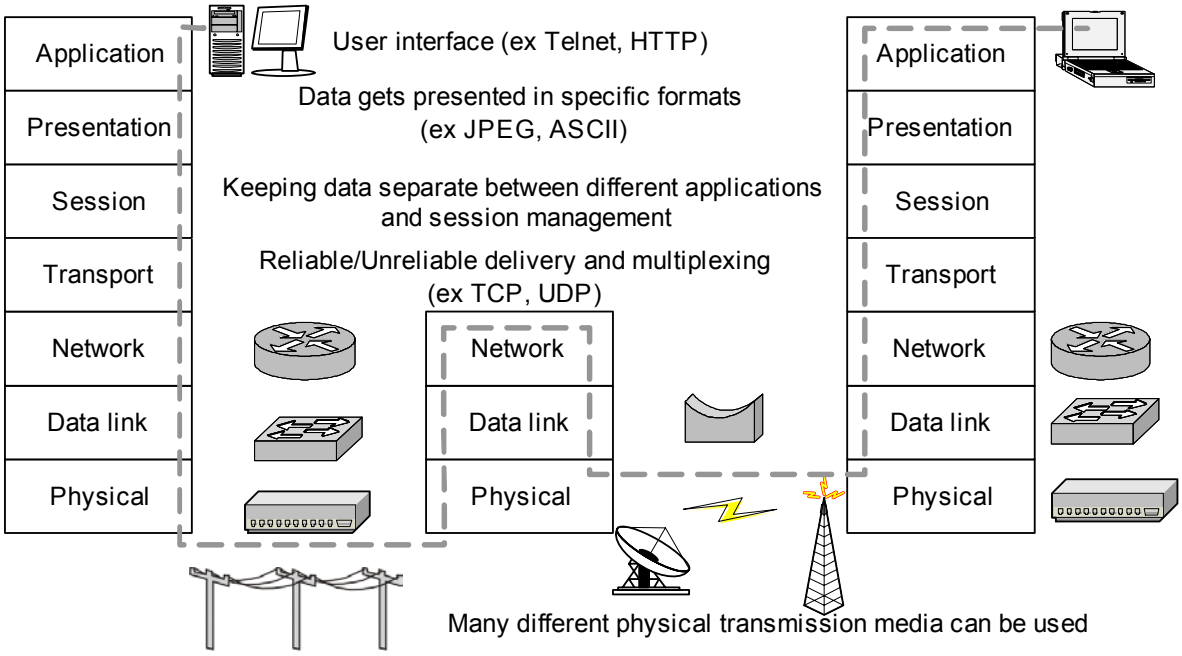


Figure 3.1 The OSI model in its practical environment

3.2.1 Application Layer

The Application Layer (Layer 7) refers to all applications providing communication services. It includes programs like WWW browsers, protocols such as HTTP or FTP, as well as other applications including voice, video and conferencing.

Relevance: This layer is important in the process of delivering QoS, for service differentiation is required based on the application. However, the DiffServ architecture, still to be discussed, distinguishes traffic based on Layer 3 header information. Consequently, it is assumed that its relevance is already incorporated in lower layer protocols.

3.2.2 Presentation Layer

The Presentation Layer (Layer 6) defines the data formats best suitable for certain types of applications. Compression and inscription are also associated with this layer with particular file formats such as ASCII, JPEG, MPEG and EBCDIC.

Relevance: The formats used in this layer are a direct result of the applications in Layer 7, and no lower level decisions are made based on these formats.

3.2.3 Session Layer

The Session Layer (Layer 5) defines how to start, maintain, and end sessions between endpoints. Examples of session layer protocols are SQL, RPC, NFS and Apple Talk ASP.

Relevance: The duration of sessions is not controllable. Performance is based on the total demand for resources, rather than discreet sessions. Admission control is beyond the scope of this dissertation. This layer is thus of no concern regarding dimensioning tasks.

3.2.4 Transport Layer

The Transport layer (Layer 4) includes the choice of protocols that may provide error recovery. Reordering and multiplexing are included as well. The most common two protocols are UDP and TCP.

Relevance: Real time and interactive applications will typically make use of UDP, whilst most other data services use TCP. With regards to dimensioning, major differences between these two protocols, are the packet size distribution (PSD) and the applications they are serving. Transport layer traffic analysis is thus a great concern.

3.2.5 Network Layer

The Network Layer (Layer 3) involves the end-to-end delivery of packets. Logical addressing and routing are performed to accomplish this. IP is the most widely used protocol.

Relevance: Part of the DiffServ functionality is implemented in the IP headers (IPv4 and IPv6). The lower layer overhead is stripped off to the Network layer in each router, as depicted in figure 3.1. A good understanding of the header and the functionality of DiffServ is

important, but packet sizes are just a mapping of Transport Layer packet sizes, with the addition of the IP header. Layer 3 traffic analysis is thus not necessary.

3.2.6 Data Link Layer

The Data Link Layer (Layer 2) transfers data across a particular link or medium. The Data Link protocols are specifically concerned with the type of Layer 1 media. Commonly used standards and protocols include IEEE 802.3/802.2, Frame Relay and ATM.

Relevance: The Data Link Layer determines lower layer packet sizes or cells (ATM). This seems relevant, but the DiffServ functionality is implemented at Layer 3 where traffic gets shaped and controlled based on layer 3 packets. When the functionality is enforced, the Physical Layer and Layer 2 overheads are neglected. A margin for this overhead has to be considered. However, Layer 2 traffic statistics do not apply.

3.2.7 Physical Layer

The Physical Layer (Layer 1) specifies the physical media. It includes different kinds of fiber, cable, electromagnetic propagation, connectors and pin configurations. Examples of Layer 1 components include FDDI, Ethernet, NRZI and B8ZS.

Relevance: The Physical Layer puts a bound on the maximum possible transmission rate. This will be of concern for the service provider with regards to fitting a maximum number of "pipes" onto a certain link, and is not the focus of this study. However, Layer 1 overhead is also neglected at layer3 dimensioning, and should thus be considered in the overhead margin.

3.2.8 Discussion

For dimensioning purposes, the following should thus be considered regarding the OSI protocol stack:

- Layer 1 and 2 overhead should be taken into account.
- The functionality of the involved Network Layer protocols should be investigated
- The traffic characteristics of the common Transport Layer protocols are very important.

- The grouping of Layer 7 applications according to their specific needs to use appropriate lower layer protocols.

3.3 OVERHEAD

The SLA describes a product which is sold as a Layer 3 service, and hence specifies the bandwidth. The access pipe provisioning is thus done such that:

$$Total\ bandwidth = \sum L3\ per\ class\ bandwidths. \tag{3.1}$$

Considering that lower layer overheads, together with Layer 3 routing and management traffic must all fit within the access pipe, one will realize that what the customer is getting, does not correspond exactly to what is sold. This section does not propose that the service is marketed differently, however it quantifies overhead to establish if it can be neglected or not. Furthermore, in the latter case it is determined how much bandwidth should be provided for these overheads.

3.3.1 Layer 1 Overhead

This is the overhead in the physical transmission media reserved for in-band alarms, signaling and management. Three of the most commonly used network infrastructures will be discussed.

ISDN

The Integrated Services Digital Network (ISDN) Physical Layer is specified by the ITU I-series and G-series documents [14]. ISDN allows voice and data to be transmitted simultaneously, using end-to-end digital connectivity. With ISDN, voice and data are carried by bearer channels (B channels) occupying a bandwidth of 64 kbps. Some switches limit B channels to a capacity of 56 kbps. A data channel (D channel) handles signaling at 16 kbps or 64 kbps, depending on the service type.

There are two basic types of ISDN services: Basic Rate Interface (BRI) and Primary Rate Interface (PRI). BRI consists of two 64 kbps B channels and one 16 kbps D channel giving a total of 144 kbps. The basic rate frame, depicted in table 3.1, is 240 bits long.

Table 3.1 Composition of ISDN basic rate frame

Sync	12 * (B1 + B2 + D)	Maintenance	Total
18 bits	216 bits	6 bits	240 bits

- The Sync field consists of 9 Quaternaries, 2 bits each, in the pattern of +3 +3 -3 -3 -3 +3 -3 +3 -3 [volt].
- (B_1 + B_2 + D) is 18 bits of data, consisting of 8 bits from the first B channel, 8 bits from the second B channel, and 2 bits of D channel data.
- The Maintenance field contains Cyclic Redundancy Check (CRC) information, block error detection flags, and "embedded operator commands" used for loop-back testing without disrupting user data.

At the prescribed data rate of 160 kbps, each frame is therefore 1.5 ms long. Each frame consists of:

- Frame overheads - 16 kbps
- D channel - 16 kbps
- 2 B channels at 64 kbps - 128 kbps

The D channel provides signalling for layers 1 to 3. An important aspect of ISDN is that data rate computations are based on the B-channels. Hence, in the SLA, the customer is quoted the actual data speed, not the pipe speed.

E1

The E1 interface provides a 2,048 Mbps access rate over coax cables. It can support up to 32 user channels, each of 64 kbps access rate. Only 30 are used as dedicated user channels. As a

consequence of the TDM methodology, each of the E1's channels is carried in one of the 32 time slots the E1's bandwidth is divided into. The concatenation of 32 consecutive time slots (TS) is named an E1 frame. The E1 frame length is 256 bits (32 TS * 8 bit each TS). The Frame rate is 8 kHz and the time slots in each frame are numbered 0 to 31:

- TS0 is dedicated for synchronization, alarms and messages (future use), unless configured differently.
- TS16 is usually used for signalling, but can carry data as well.
- TS1-TS15 and TS17-TS31 are used for carrying user data.

The definite payload bandwidth is thus 1.920 Mbps, 93.8% of the total bandwidth which corresponds to a worst case 6.2% overhead.

SONET/SDH

SONET and SDH are a set of related standards for synchronous data transmission over fiber optic networks. SONET is the acronym for Synchronous Optical Network, and SDH stands for Synchronous Digital Hierarchy. SONET is the United States version of the standard, and is published by the American National Standards Institute (ANSI). SDH is the international version of the standard, published by the International Telecommunications Union (ITU). Table 3.2 lists the hierarchy of the most common SONET/SDH data rates:

Table 3.2 SONET/SDH data rates and associated overheads.

Optical Level	Electrical Level	Line Rate (Mbps)	Payload Rate (Mbps)	Overhead Rate (Mbps)	SDH Equivalent
OC-1	STS-1	51.840	50.112	1.728	-
OC-3	STS-3	155.520	150.336	5.184	STM-1
OC-12	STS-12	622.080	601.344	20.736	STM-4
OC-48	STS-48	2488.320	2405.376	82.944	STM-16
OC-192	STS-192	9953.280	9621.504	331.776	STM-64

OC-768	STS-768	39813.120	38486.016	1327.104	STM-256

In each of these cases, the payload is 96.7% of the total bandwidth, which corresponds to a total overhead of 3.3%

Discussion

Layer 1 overhead thus varies between 0% and 6.2%. Research in [15] shows that there is a definite increase in the deployment of SDH giving an overhead of 3.3%.

3.3.2 Layer 2 Overhead

The general Layer 2 overheads are Medium Access Control (MAC) framing and Logical Link Control (LLC) headers such as SNAP, FRF.12, ML and PPP. The overhead of Frame Relay and Asynchronous Transfer Mode (ATM), the two most commonly used Layer 2 protocols, will be investigated.

Frame Relay

A Frame Relay network implements the *Link Access Procedure for Frame mode bearer services* (LAPF) core protocol, to perform the needed data link control functions. The frame format is shown in figure 3.2.

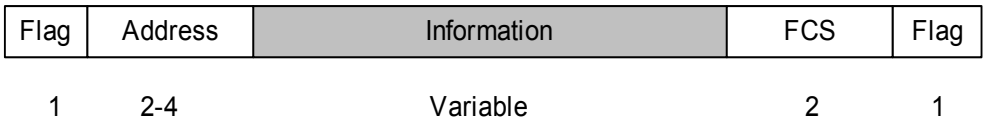


Figure 3.2 LAPF – core frame format

The LAPF frame format shown in figure 3.2 corresponds to an overhead of 6 to 8 bytes (48-64 bits). The number of data bytes varies and thus it is not possible to derive a definite percentage overhead. However, an average overhead can be determined by using a packet size distribution:

Assume that packets arrive according to a distribution a, b, c with probabilities α, β, γ respectively. The average packet size is hence

$$\rho = \alpha \times a + \beta \times b + \gamma \times c. \tag{3.2}$$

The average offered load (in bps) is thus

$$\bar{L} = \bar{\rho} \times \xi(a, b, c), \tag{3.3}$$

where ξ(a, b, c) is the average packets per second. In the above construction, the packets per second of a, b, c are all identical; although their bit rates vary. The aggregate load can be thought of as bit streams at respective rates a/ξ(a, b, c), b/ξ(a, b, c) and c/ξ(a, b, c), which are sampled with probabilities α, β and γ. Each packet has the same overhead of size Δ (bits per packet), and the average head load in bps is hence

$$\bar{H} = \xi(a, b, c) \times \Delta. \tag{3.4}$$

Thus, the average percentage overhead is

$$\bar{\theta} = \frac{\bar{H}}{\bar{L}} = \frac{\Delta}{\bar{\rho}}. \tag{3.5}$$

The purpose of finding θ is that it can be taken into account during dimensioning. As a result of equation 3.4, Δ is known and $\bar{\rho}$ can be obtained through empirical results. Finding θ can also help to establish the limiting[2] rate. It can be set to $\bar{L} \times (100+\bar{\theta})$ %, effectively providing for the overhead as well.

ATM

ATM uses fixed-size cells, consisting of a 5-octet header and a 48-octet information field. The overhead is thus 5/53 or equivalently 9.4%. The cost of such a high overhead implies a gain on

[2] Limiting is performed to enforce arriving traffic to comply with the traffic profile described in the SLA. More detail is provided later in this chapter.

other levels; as it reduces queuing delays for high priority classes and it is easier to implement switching mechanisms in hardware for fixed cell sizes.

Discussion

Layer 2 overhead is thus variable in the case of Frame Relay, and 9.4% in the case of ATM. The combination of ATM (layer 2) and SDH (layer 1) results in an overhead of 12.5%. The effect of these overheads will be analyzed after simulations have been performed, and included in the total bandwidth budget as applicable.

3.4 RELEVANT PROTOCOLS

Understanding the functionality of certain network architectures requires knowledge of the protocols involved. In this section, IP as a Layer 3 protocol and Multi Protocol Label Switching (MPLS) will be discussed in terms of their relevance regarding DiffServ.

3.4.1 IP

The Internet Protocol version 4 (IPv4) is currently the most prominent Network layer protocol together with its successor, IP version 6 (IPv6). The IPv4 packet header is shown in figure 3.3.

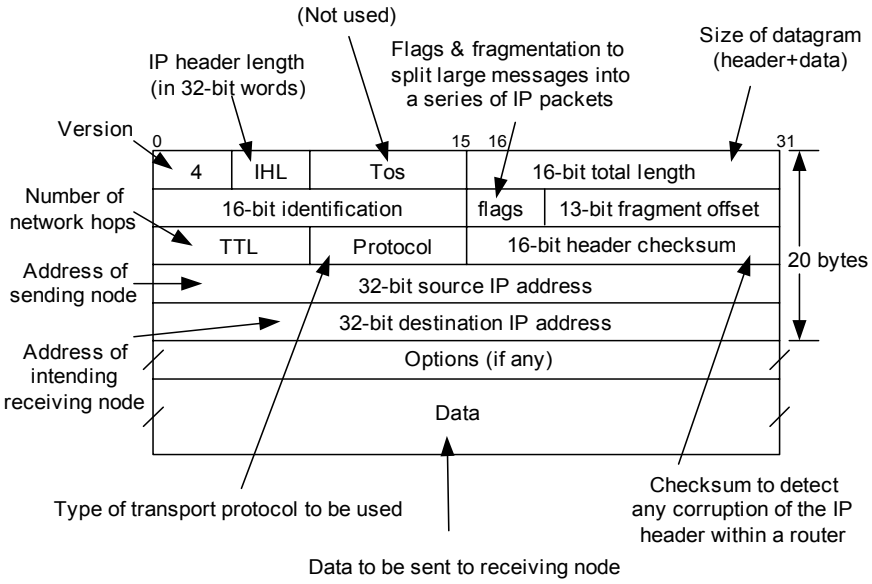


Figure 3.3 The IPv4 packet header

DiffServ forwarding decisions are made based on the Network layer packet header. The only field directly relating to DiffServ is the ToS byte. As shown in figure 3.3, the ToS field is traditionally not used. To implement DiffServ, a specific code known as the DSCP, is used in the ToS byte. It is normally set to 0, but the use of the DSCP indicates particular QoS needs from the network. The DSCP defines the service class. In IPv6 the ToS byte is renamed to the DiffServ field also known as the priority byte, shown in figure 3.4.

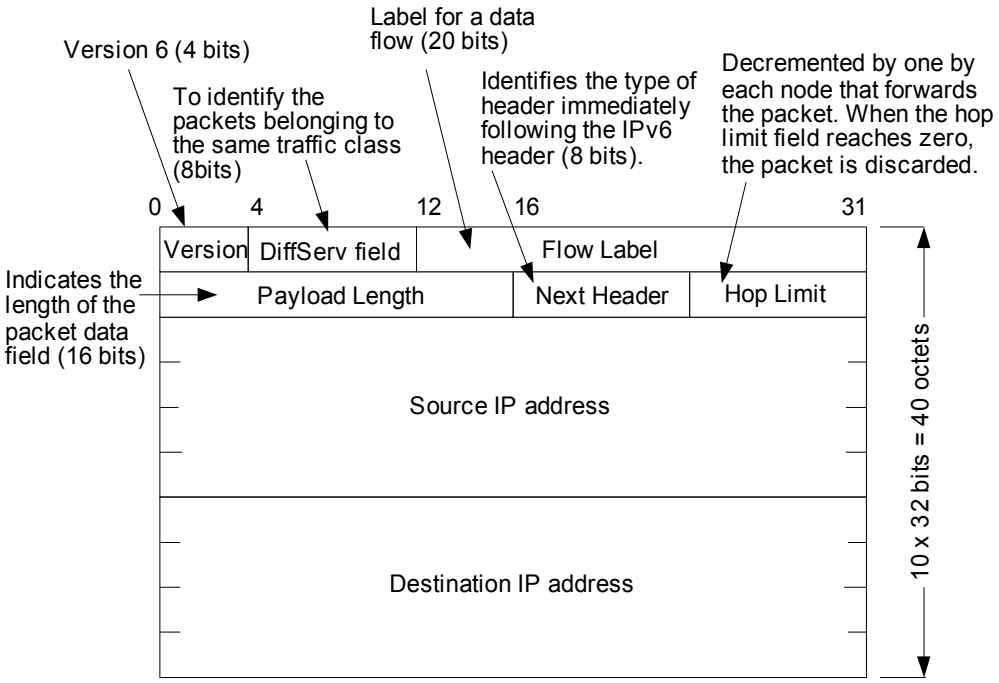


Figure 3.4 The IPv6 Packet header

The IPv6 header was designed to be less complex than the Ipv4 header; an attempt was made to offload the options to post IP headers. The number of bits in the IP header is still 40 bytes, as opposed to 20 bytes in Ipv4, due to the larger address space.

The DSCP

The DSCP determines the packet's *Per-Hop-Behaviour* (PHB) at each node throughout the DiffServ domain. The PHB can be described as the externally observable forwarding treatment applied at DiffServ compliant nodes to behaviour aggregates [16]. A *Behaviour Aggregate*

(BA) is a collection of packets with the same code point crossing a link in a particular direction.

The ToS byte in version 4 and the DiffServ field in version 6 are both 8 bits long. The particular bits defined in each of these bytes are depicted in the following figures.

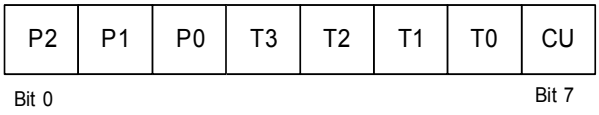


Figure 3.5 The ToS byte (IPv4)

Bits P2 to P0 are the 3 IP precedence bits, bits T3 to T0 are the 4 ToS bits, and CU is one currently unused bit.

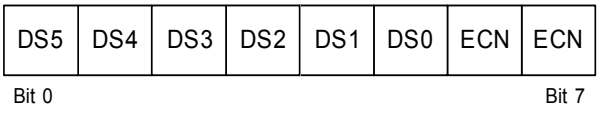


Figure 3.6 The DiffServ field (IPv6)

Bits DS5 to DS0 are the 6 DSCP bits and the ECN bits are Early Congestion Notification.

The first 3 bits of the DiffServ field (DS5 to DS3) are the class selector bits and correspond to the ToS precedence bits. The rest of the bits do not correspond directly. DS2 to DS0 in the DiffServ field represent delay, throughput and reliability requirements, while T3 to T1 in the ToS byte represent the drop probability. Table 3.3 illustrates how these bits are used to differentiate between classes.

Table 3.3 Bit 0–bit 2 of the ToS byte (P2-P0) and the DiffServ field (DS5-DS3).

Bits 0-2	Precedence
111	Network control: Precedence 7. Link layer and routing protocol keep alive.
110	Inter-network control: Precedence 6. IP routing protocols
101	CRITIC/ECP: Precedence 5. Express Forwarding (EF)
100	Flash override: Precedence 4. Class 4.
011	Flash: Precedence 3 Class 3.
010	Immediate: Precedence 2. Class 2.
001	Priority: Precedence 1. Class 1
000	Routine: Precedence 0. Best effort

The first 3 bits are thus used to indicate the class. DiffServ nodes prioritize traffic by class first. It would then differentiate and prioritize same-class traffic based on bits 3 to 5. This further differentiation is not the same for the ToS byte and the DiffServ node as in the case of the first 3 bits. Table 3.4 and 3.5 indicate these differences.

Table 3.4 Bit 3–bit 5 of the ToS byte (T3-T1)

Bits 3-5	Drop probability
010	Low
100	Medium
110	High

Table 3.5 Bit 3-bit 5 of the DiffServ field (DS2-DS0)

Bits 3-5	Parameter description
3	Delay [D]: 0=Normal, 1=Low
4	Throughput [T]: 0=Normal, 1=High
5	Reliability [R]: 0=Normal, 1=High

The DiffServ standard does not specify a precise definition of drop probabilities, delay, throughput or reliability description. Not all devices recognize the bit 3 to 5 DiffServ settings. When they are, the same PHB is not guaranteed. The technology still needs to be refined before this fine detail traffic specification could be used.

With reference to table 3.3, Classes 1 to 4 and Best Effort, belong to the Assured Forwarding (AF) PHB group [17]. The way that forwarding decisions are made for these classes can be expected to be the same, except for different weights allocated to each of the classes. The Expedited Forwarding (EF) PHB is defined in [18]. It is used to build low latency, low jitter, low loss, assured bandwidth and end-to-end service through DiffServ domains.

Regarding protocols and header information, this section summarized how a small number of bits in the IP header are used to differentiate between traffic classes. Section 7 introduces the physical node architecture to realize the described diverse forwarding behaviour. Another very important protocol in terms of DiffServ is Multi Protocol Label Switching (MPLS). MPLS was designed to ensure fast packet switching in the core network and cater for scalability and migration.

3.4.2 MPLS

MPLS is responsible for fast switching and operates over multiple layers in the protocol stack as depicted in figure 3.6.

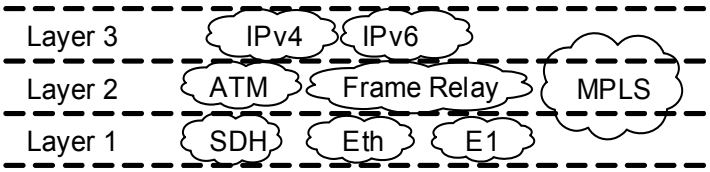


Figure 3.7 MPLS's multi-layered functionality

MPLS is called multi-protocol because it encapsulates IP, ATM and Frame Relay (layer 2 and 3 protocols). It allows packets to be forwarded at both Layer 2 and Layer 3, establishing virtual connectivity between sites in a way that emulates the connection oriented nature that

has made ATM and Frame Relay important technologies. This is done by encapsulating Frame Relay, ATM or IP over MPLS as shown in figure 3.8.

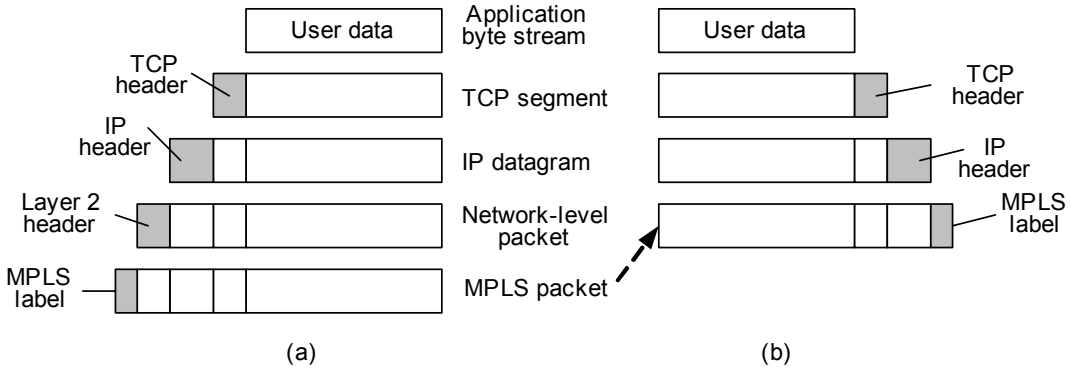
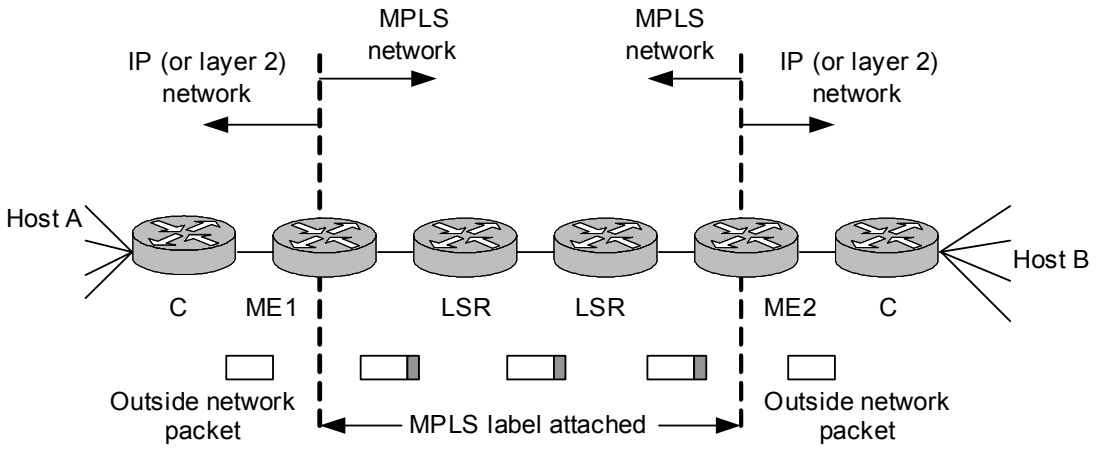


Figure 3.8 MPLS encapsulating layer 2 protocols (a) as well as layer 3 (b)

A label is a header used by a Label Switched Router (LSR) to forward packets. The header format depends upon network characteristics. In router networks, the label is a separate 32 bit header. In ATM networks, the label is placed into the Virtual Path Identifier/Virtual Channel Identifier (VPI/VCI) cell header. In the core, LSRs read only the label, not the network layer packet header. One key to the scalability of MPLS is that labels have only local significance between two devices that are communicating.

Figure 3.9 shows an MPLS domain and the process of packet labelling. Packets arrive as IP packets at ME1, the MPLS edge router, also known as the ingress label switching router. At the ingress router, the IP precedence bits are copied into three bits of the MPLS label that comprise an experimental field. ME1 transmits the packets as MPLS packets. Within the MPLS network, the queuing mechanism doesn't look at any data in the IP header, as the packets are MPLS packets. The packets remain MPLS packets until they arrive at ME2, the egress LSR. ME2 removes the label from each packet and forwards the packets as IP packets.



C – Customer Edge Router,

ME – MPLS Edge Router,

LSR – MPLS core Label Switched Router.

Figure 3.9 The MPLS domain

MPLS is known for its good switching time relative to other layer 2 and 3 protocols. The reason is that non-MPLS networks use the destination address along with a look-up table to determine an output port, and layer 2 headers get rewritten at each hop. In an MPLS network, LSRs use labels instead of addresses. Routers advertise shortest paths and the label designates the output port at each hop. MPLS switching is significantly faster than the legacy routing processes.

MPLS offers a credible migration path from Layer 2 solutions, providing what could be described as a bridge between traditional Layer 2 technologies and full IP at Layer 3. Enterprises can migrate seamlessly to IP products when they feel it is appropriate. In this way service providers can continue to support legacy, revenue-generating data services to protect existing customer relationships and to fund new network developments. In addition, it promises superior scalability, flexibility and cost efficiencies. For these reasons, it is widely claimed in literature that MPLS will supersede ATM within the next five years. It is expected to be readily adopted as networks begin to carry larger volumes and different mixtures of traffic.

3.5 DIFFSERV

The DiffServ domain is limited to portions of the network where the necessary functionality is implemented on network nodes. Three types of nodes, or routers, exist in a DiffServ domain:

- CE – Customer Edge Routers. These are customer propriety, but are mostly managed by the service provider.
- PE – Provider Edge Routers. These routers are linked to both provider routers and customer routers.
- P- Provider Core Routers. These routers are only linked to other provider routers.

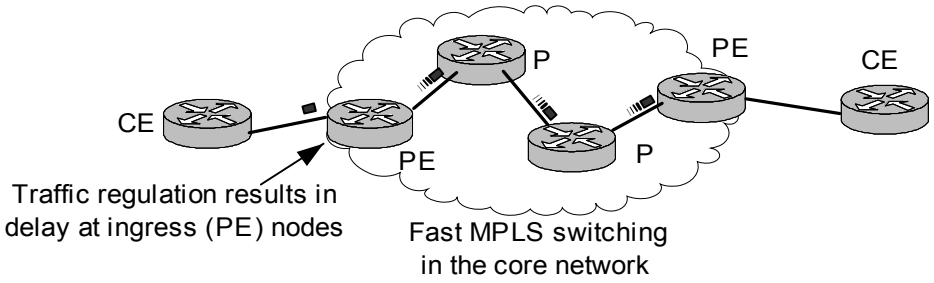


Figure 3.10 The DiffServ domain

The DiffServ domain is illustrated in figure 3.10. The main reason for the distinction in the type of nodes is for the distribution of the functions that need to be performed. The aim is to move the functionality resulting in degradation of QoS to the edges of the network. Once a packet enters the core, switching is performed at high speed without the addition of a significant delay or loss.

The DiffServ domain does not inherently imply an MPLS core. MPLS is normally used in the core to improve edge-to-edge delivery time. MPLS and DiffServ are orthogonal, in the sense that DiffServ operates on per-class aggregates, whereas MPLS operates on per-flow aggregates. MPLS is thus used to direct traffic flows based on destination and source characteristics, whilst DiffServ is used to manipulate traffic based on class priority. MPLS itself provides no QoS functionality, other than to assist in effective DiffServ deployment.

As data traverses the DiffServ-enabled network, various tasks are performed on the packets, influencing their movement. Figure 3.11 indicates the nodes where these tasks are carried out as a function of the data flow direction.

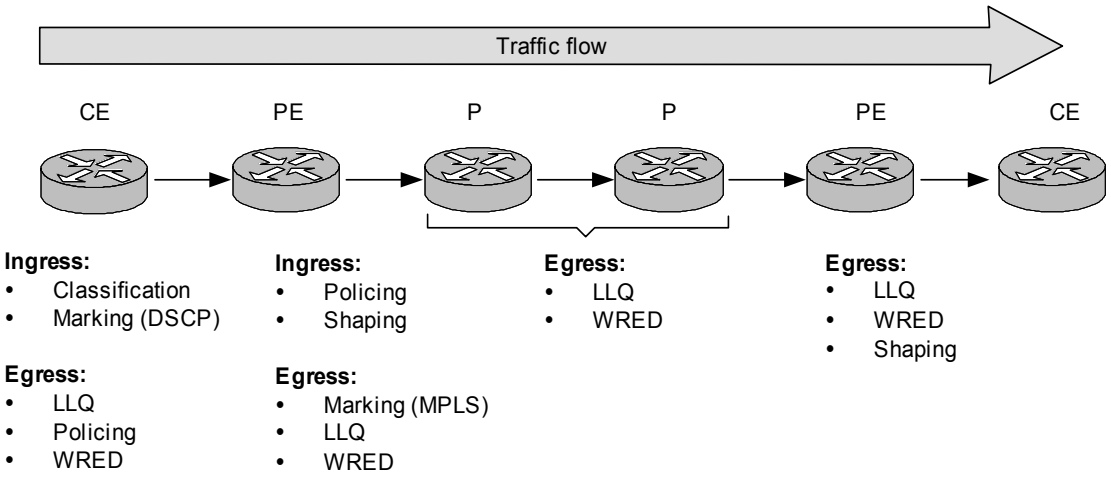


Figure 3.11 The functions being performed at the distinctive DiffServ nodes

A detailed description of these functions and their effect on QoS is presented in section 3.7. First however, it is necessary to introduce the traffic classes, as the functions differ for the various traffic classes.

3.6 TRAFFIC CLASSES

Traffic classes are created in order to establish a way of differentiating between various aggregates of traffic. Vendors and standards bodies define their own traffic classes known as Classes of Service (CoS). The number of specified classes is not consistent, but there is generally a split between *Guaranteed Services* (GS) and *Best Effort* (BE). The IETF only defines these classes shown in table 3.6.

Table 3.6 The IEFT traffic classes

Traffic Class	Description	Applications
GS	Intolerant to delay: Worst case delay guarantees.	Network gaming Voice and video

	Tolerant to delay: Nominal delay, can tolerate occasional deviation and loss.	
BE	Interactive Burst Interactive Bulk Asynchronous Bulk	Messaging, email FTP Netnews, spam

CISCO defines the traffic classes depicted in table 3.7 [6].

Table 3.7 DiffServ traffic classes defined by CISCO

Traffic Class	Description	Applications
Premium	To deliver minimum delay. Reserved for voice only. Maximum of 500kbps during congestion.	Voice
Gold	Preference over silver. Minimum guarantee: 35% of total link bandwidth	TACACS[3]
Silver	Preference over bronze. Minimum guarantee: 25% of total link bandwidth	Telnet SMTP FTP
Bronze	15% of total link bandwidth	HTTP

Telkom South Africa Limited (SA Ltd) has made a substantial effort to offer a service which considers specific customer requirements. The four classes defined by Telkom are shown in table 3.8 [19].

Table 3.8 DiffServ traffic classes defined by Telkom SA Ltd

Traffic Class	Description	Applications
Real Time (RT)	Optimized for voice and other MM applications with low delay	Voice, MM Conference Digital

[3] TACACS – Terminal Access Controller Access Control System, an access control protocol described in [20].

	requirements	Video
Interactive Business (IB)	Suitable for interactive data applications requiring low loss and low latency.	SAP, Telnet, Oracle, SAP
Bulk Business (BB)	Provides managed loss and delay for the majority of core business applications. Priority and Non-Priority differentiation is made: BB1 and BB2.	File Printing/Sharing, GroupWise, Lotus Notes, MS Exchange
General Data (GD)	Delay and throughput average with very long sample periods (one month) are guaranteed. Priority and Non-Priority differentiation is made: GD1 and GD2.	FTP, Email, WWW applications

The remainder of this dissertation is based on the traffic classes defined by Telkom.

3.7 NODE ARCHITECTURE

A visualization of the DiffServ node architecture is depicted in figure 3.12. The illustrated node architecture is described in terms of its distinctive components hereafter.

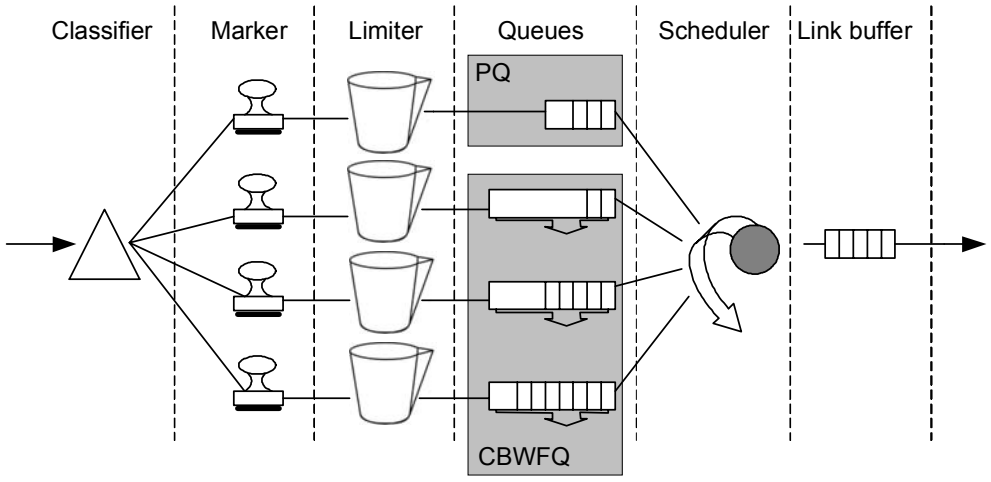


Figure 3.12 DiffServ node architecture

3.7.1 Classifier

Packet classifiers select packets in a traffic stream based on the content of some portion of the packet header. Two types of classifiers are defined [21]:

The Behaviour Aggregate (BA) classifier classifies packets based on the DSCP only.
The Multi-Field (MF) classifier selects packets based on the value of a combination of one or more header fields. Such header fields include the source address, destination address, DiffServ field, protocol ID, source port and destination port numbers, and other information such as the incoming interface.

Classifiers are configured by management procedures in accordance with the applicable Terminal Control Area (TCA).

Classifiers are used to "steer" packets by matching a specified rule to an element of a traffic conditioner for further processing. With reference to figure 3.12, this can be interpreted visually as each packet arriving at the classifier, is forwarded to one of the four links according to the class to which it belongs to.

The classifier additionally serves as a defence mechanism against theft and denial-of-service attacks. This form of security is based on modified DSCP values and code points to which the traffic is not entitled. Such unauthorized traffic is discarded, because it constitutes a violation of the applicable TCA(s) and/or service provisioning policy.

3.7.2 Marker

Packet markers set the DiffServ field of a packet to a particular code point, adding the marked packet to a particular BA. The marker may be configured to mark all packets that are routed to it to a single code point, or may be configured to mark a packet to one of a set of code points used to select a PHB in a PHB-group. When the marker changes the code point in a packet, it is said to have "re-marked" the packet.

3.7.3 Limiter

Two types of limiters are commonly used: polices and shapers. Polices (or droppers) are used for delay-sensitive traffic, and hence are implemented for the RT class. Shapers are used for traffic that can tolerate some delay, and hence are used for the rest of the data classes. Before the functionality of these limiters is discussed, traffic profiles and the token bucket metaphor are described.

Traffic Profiles

User traffic must conform to a traffic profile defined in the SLA in order to receive the guaranteed QoS. The traffic profile is defined by parameters such as the permitted transmission rate and the burstiness of the traffic. Traffic conforming to the specified profile is referred to as in-profile, or in-bound traffic. Likewise, the non-conforming traffic is known as the out-of-profile, or out-of-bound traffic. In-bound traffic is forwarded on its intended route according to the forwarding policy. Out-of-bound traffic gets policed or shaped. A token bucket, or leaky bucket, is used to establish whether incoming traffic is in-bound or out-of-bound.

Token Buckets

Simply stated, both shaping and policing use token buckets. The token bucket is a metaphor used to visualize how rate limiting works:

- Tokens are put into the bucket at a certain rate, known as the Committed Information Rate (CIR).
- Each token gives permission for the source to send a certain number of bits into the network.
- To send a packet, the traffic regulator must be able to remove a number of tokens equal in representation to the packet size, from the bucket.
- If there are not enough tokens in the bucket to be able to send a packet, the packet either waits until the bucket has enough tokens, in the case of a shaper, or the packet is discarded or marked down, in the case of a police.

- The bucket itself has a specified capacity. If the bucket is filled to its capacity, newly arriving tokens are discarded and are not available to future packets. Thus, at any given time, the largest burst a source can send into the network is roughly proportional to the size of the bucket. A token bucket permits burstiness, but at the same time restricts it.

Figure 3.13 illustrates the token bucket metaphor.

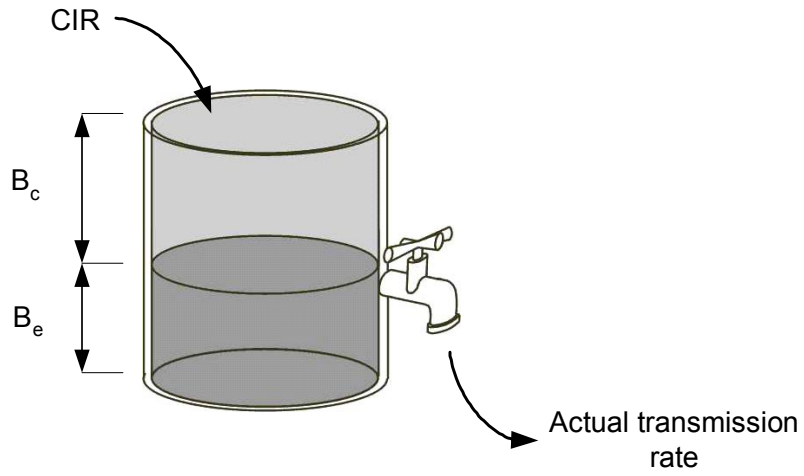


Figure 3.13 Visual representation of the token bucket metaphor

B_c represents the committed burst, while B_e is the excess burst capacity, and *CIR* stands for Committed Information Rate. Tokens are added to the bucket at the *CIR* once every window period. The policing window is defined as:

$$T_c = B_c / CIR \tag{3.6}$$

Once every window period, B_c tokens are added to the bucket. Any arriving packet gets transmitted as long as the state of the bucket exceeds B_e with sufficient amount of tokens for the concerned packet. When a packet arrives and there are only sufficient tokens in the bucket such that half the packet could be transmitted, the packet is simply delayed until the bucket fills up sufficiently, in the case of a shaper. However, delay is not an option in the case of a policer, thus the packet should be dropped. In order to reduce packet dropping in policers, bursts are allowed to borrow from the excess burst capacity, B_e. Token borrowing must conform to the following rules:

- If the packet attempting to borrow tokens is the *i'th* since the last time a packet was dropped, calculate the compound debt C_d as

$$C_d = \sum_{j=1}^{i} \sum_{k=1}^{j} \textit{tokens borrowed by packet k.} \tag{3.7}$$

- Hence, compound debt grows exponentially as more packets borrow tokens.
- If $C_d \leq B_e$, the packet gets admitted,
- Else the packet *i* is dropped, and C_d is set to zero,
- The actual debt is the sum of the tokens borrowed by packet *k*, which must now be paid back. If the next packet arrives before the actual debt is amortized, its compound debt is set to the actual debt value. The packet may be transmitted or dropped based on compound debt.
- Even if the actual debt value reaches B_e such that a packet must be dropped, a few time intervals later, the packet may be sent without the actual debt being paid off. In this case, traffic may always fall within the B_c and B_e margin.

A token bucket itself has no discard or priority policy. Priority is determined by the scheduling discipline as well as the queue management. The discarding policy is determined by the shaping and policing functions.

Shapers

Shapers delay, or buffer, some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets. In a shaper, the token bucket is incremented at timed intervals, using a bits-per-second (bps) value. The following formula is used:

$$T_c = B_c / CIR \tag{3.8}$$

The value of T_c defines the time interval during which B_c bits are sent in order to maintain the average rate of the CIR. A typical value used for T_c is 125 ms [22]. This value allows relatively big bursts, although it still has a good smoothing effect. B_e is thus not necessary for shapers, for packets get delayed. Defining a value for B_c has the same effect as increasing the B_c value.

Most data traversing the modern day networks can tolerate some delay. The shaper is thus used as a limiting device for most of the traffic classes. A shaper cannot be used for delay sensitive data, since packet delays will result in the decrease of the QoS. Policers are used for delay sensitive data.

Policers

Policers discard instead of delay. When out-of-band packets arrive at an ingress node, some or all of the packets in the traffic stream are discarded in order to bring the stream into compliance with a traffic profile. This process is known as "policing" the stream. The result of out-of-band traffic will be packet loss, rather than packet delay, which is favourable to real-time applications.

Token buckets are also used in policers, but the tokens can be added to the bucket in different ways. The first method is the same as the method used for shaping, except that the policing window, T_c has a much shorter duration of 10ms [22], due to real-time traffic's intolerance to delay. The second method adds tokens continuously, relative to the packet flow. The following formula applies;

$$\frac{T_p * R_p}{8}, \tag{3.9}$$

where T_p is the time between arriving packets, R_p is the police rate and the 8 accounts for the number of bits in a byte. In this way, each packet is evaluated individually for conformance to the traffic profile. Opinions, regarding the use of B_e in policing, differ in literature and among

vendors Cisco [23] states that B_e should be zero particularly because of strict delay requirements. Telkom best practice recommendations, advises that B_e should be given a value in order to reduce packet dropping. Both of these methods will be considered in the practical simulations.

Policing Versus Shaping

The following diagrams illustrate the key differences. Traffic policing propagates bursts. When the traffic rate reaches the configured maximum rate, excess traffic is dropped, or remarked. This results in an output rate that appears as a saw-tooth with crests and troughs. In contrast to policing, traffic shaping retains excess packets in a queue, which then schedules the excess for transmissions following over increments of time. The result of traffic shaping is a smoothed packet output rate as depicted in figure 3.14.

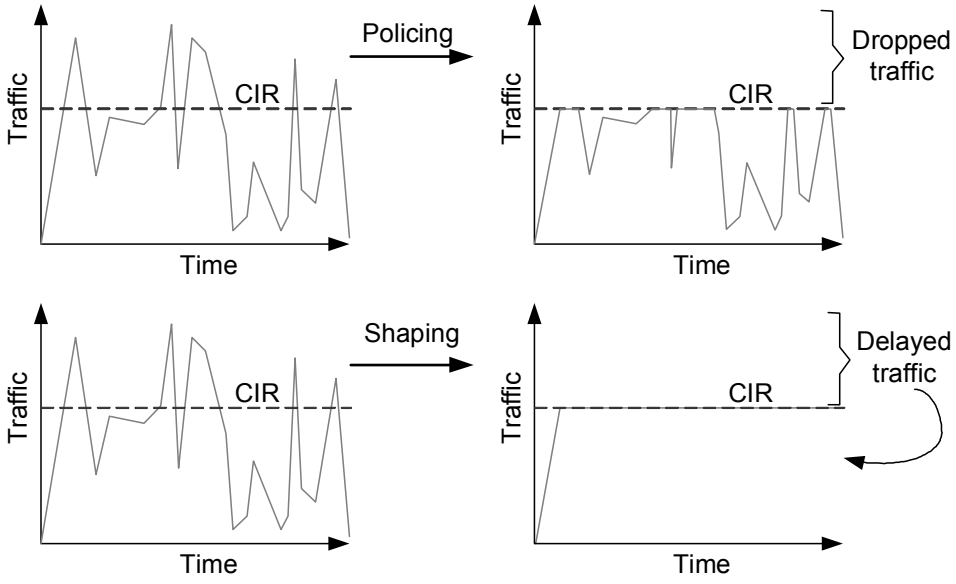


Figure 3.14 Effects of policing and shaping on a bursty traffic source

Policing and shaping each have its own favourable characteristics, applications and place in practice. The following table lists the differences between shaping and policing [23]

Table 3.9 Comparison of policing and shaping

Attribute	Shaping	Policing
Objective	Buffer and queue excess packets above the committed rates.	Drop (or remark) excess packets above the committed rates. Does not buffer[4]
Token Refresh Rate	Incremented at the start of a time interval. (Minimum number of intervals is required.)	Continuous based on formula: 1 / committed information rate
Token Values	Configured in bits per second.	Configured in bytes.
Applicable on Inbound	No	Yes
Applicable on Outbound	Yes	Yes
Bursts	Controls bursts by smoothing the output rate over at least eight time intervals. Uses a leaky bucket to delay traffic, which achieves a smoothing effect.	Propagates bursts. Does no smoothing.
Advantages	Less likely to drop excess packets since excess packets are buffered. (Buffers packets up to the length of the queue. Drops may occur if excess traffic is sustained at high rates.) Typically avoids	Controls the output rate through packet drops. Avoids delays due to queuing.

[4] Although policing does not apply buffering, a configured queuing mechanism applies to "conformed" packets that may need to be queued while waiting to be serialized at the physical interface.

	retransmissions due to dropped packets.	
Disadvantages	Can introduce delay due to queuing, particularly deep queues.	Drops excess packets (when configured), throttling TCP window sizes and reducing the overall output rate of affected traffic streams. Overly aggressive burst sizes may lead to excess packet drops and throttle the overall output rate, particularly with TCP-based flows.

3.7.4 Queues and Schedulers

Queuing and scheduling goes hand-in-hand. A queue is essentially the buffer(s) being served by a scheduler. Inside the queue is normally a first-in, first-out (FIFO) operation, which operates in the same way when people naturally organize themselves at shopping queues. This is also known as first-come, first-serve (FCFS).

It is actually the scheduler that controls the queues, which determines the queuing discipline. A scheduling discipline is an algorithm implemented by a scheduler, which determines the order in which incoming packets are fed to the output buffer(s) or link(s) [24]. Traditional IP routers used FIFO since its inherent simplicity makes it fast. However, in order to implement QoS a more complex algorithm must be found. In the DiffServ node, where several packet streams are arriving simultaneously, priority has to be given to some streams. Priority implies that a flow is given lower delay, or a higher data rate. A flow is given priority at the expense of other flows [25]. This is known as the Kleinrock's conservation law which states that:

$$\sum_{n=1}^{N} \rho_n q_n = C \tag{3.10}$$

$\rho_n = \lambda_n . \mu_n =$ mean link utilization

$q_n =$ mean packet waiting time due to scheduler

$\lambda_n =$ mean packet arrival rate

$\mu_n =$ mean packet service time

A scheduling discipline is called work conserving when packets are served for as long as there are time slots available. A non-work conserving scheduler on the other hand, wastes bandwidth by allowing packets to wait in their queues for the appropriate service time. Work conserving disciplines use a sorted priority queue since the scheduler has the flexibility to perform functions of delay bounding, bandwidth allocation and adjusting for traffic pattern distortions. In general, it results in much more effective bandwidth utilization. Non-work conserving scheduling disciplines may be used in order to maintain a known traffic distribution along a path, which makes network analysis easier. Simulation results presented in chapter 5 are based on a work conserving scheduler.

The main objectives of scheduling algorithms in modern networks are to:

- Maximize router/switch throughput utilization,
- Minimize mean packet delay,
- Minimize packet loss resulting in buffer overflow, and to
- Minimize strict QoS requirements in accordance with diverse data classes.

For complete surveys of queue scheduling disciplines, refer to [7] and [24]. A performance analysis of queue scheduling mechanisms for EF PHB and AF PHB in DiffServ networks are documented in [26]. [27] presents the round-robin-priority queuing (RRPQ) discipline, with delay bounds without per-flow queuing. Another proposal (EBVND) [28] analyses and improves quality guarantees theoretically. Similar to these results, numerous queue scheduling disciplines are defined in literature, where analyses are being performed and documented, and

new proposals are being made. In this work, only relevant disciplines with regard to the DiffServ node architecture will be reviewed.

FIFO

The FIFO algorithm forms the foundation of the other scheduling disciplines [29]. Extensive work by [30], [31], [32] and [33] concluded that the worst-case delay is found by dividing the buffer size by the link speed. This result is also used in the simulations presented in chapter 5. The distinctive queue itself is FIFO, but the combined scheduling discipline is a more complex mechanism, still to be discussed.

Priority Queuing

For a priority scheduler, several FIFO queues are piled on top of each other in decreasing order of priority. Priority queuing (PQ) provides the most simplistic method of supporting differentiated service classes. In classic PQ, packets are first classified and then placed into different priority queues. Packets are scheduled from a particular queue, only if all queues of higher priority are empty. Within each of the priority queues, packets are scheduled in a FIFO order.

Class Based Weighted Fair Queuing

For Class Based Weighted Fair Queuing (CBWFQ), the weight for a packet belonging to a specific class is derived from the bandwidth assigned to the class when it was configured. Therefore, the bandwidth assigned to a class determines the order in which packets are sent. All packets receive weight based equal treatment; no class of packets may be granted strict priority. This scheme poses problems for voice traffic that is largely intolerant of delay, especially variation in delay. For voice traffic, variations in delay introduce irregularities of transmission, manifesting as jitter in a conversation.

Low Latency Queuing

Low Latency Queuing (LLQ) was proposed by CISCO [34], in particular to address the problem that CBWFQ was facing with regards to delay sensitive traffic. LLQ adds strict priority queuing to CBWFQ. It can be described as a QoS policy that enables users to define traffic classes in terms of customer-defined matched criteria. Delay guarantees for real-time applications to be granted, due to the presence of the priority queue.

Under CBWFQ, once classes have been specified, network administrators can then apply parameters such as bandwidth and queue-limits to these classes. Even with a large weight allocated under CBWFQ, delay sensitive data cannot be guaranteed adequate performance at all times.

LLQ goes a step further, by offering better treatment and a strict priority queue (PQ) for delay-sensitive data, such as voice, while providing weighted fair queues to other traffic classes. When the LLQ feature is used, delay-sensitive data will first be serviced and then sent, before packets in alternative queues are serviced, thus giving it preferential treatment over other traffic. This lowers the delay for voice packets, which is very important for voice quality, especially on slow data links.

The LLQ feature reduces jitter in voice conversations. LLQ is recommended on all Virtual Circuits (VC) where voice, video or any real-time traffic are involved. Packets must be marked with adequate precedence[5] to be queued in the priority queue. Packets that do not meet any of these criteria will be sent to one of the other classes where they will be given a fair queuing treatment.

Due to the nature of this high priority queue, it is important to police the traffic entering this queue. Without policing, misbehaving traffic sources would be able to dominate the server, resulting in poor performance for the rest of the classes.

[5] The DiffSev Code Point (DSCP) or the IPv4 ToS byte precedence bits are used to indicate the need for high priority.

3.7.5 Output Link Buffer

The output link buffer is a single FIFO queue, with the purpose of allowing the scheduler to perform its tasks without having to wait for packets being serialized on the output link.

3.8 RESOURCE DIMENSIONING

Once the choice of architecture and all its accompanying mechanisms and disciplines have been established, it is still left to be determined how many resources should be dimensioned for the concerned setup. Regarding the DiffServ node architecture, it basically amounts to 3 attributes that have to be quantified, namely:

- Bandwidth,
- Token bucket parameters, and
- Buffer requirements.

The bandwidth allocated for a service is the most crucial, for it has a direct impact on the cost and the QoS. The token bucket parameters do not have a cost implication, but are important for regulating incoming traffic to comply with the predefined traffic profile. Buffers have a small cost implication and can easily be derived from the token bucket parameters once these have been established. As a result, this work only considers the dimensioning of bandwidth and token bucket parameters.

3.8.1 Bandwidth

To make the best use of a particular network infrastructure, seen from the service provider, as well as the client point of view, three basic responsibilities must be comprised:

- For a fixed network infrastructure with specified bandwidth and buffer space, the service provider has to sell as much as possible of these network resources, in order to justify the cost of the infrastructure.
- The onus lies with customers to define the bandwidth to accurately suit their needs.

- The service provider has the responsibility to provide sufficient bandwidth in order to comply with QoS guarantees while maximizing utilization, which in turn results in better profit.

This study is concerned with the third point. Mainly two types of bandwidth allocation are commonly encountered: Dynamic and static bandwidth allocation. Bandwidth allocation refers to two different concepts in literature. The first is the actual scheduling discipline that shares bandwidth between classes (see figure 3.15). A well-known static discipline is *Round Robin Queuing* (RRQ), whereas *Priority Queuing* (PQ) is a typical dynamic discipline. Static schedulers determine which queue will be served, irrespective of the incoming traffic streams. If the scheduler is affected in any way by any of the incoming traffic streams, it is considered a dynamic scheduler. This dissertation refers to this concept as static or dynamic *bandwidth sharing*, rather than bandwidth allocation.

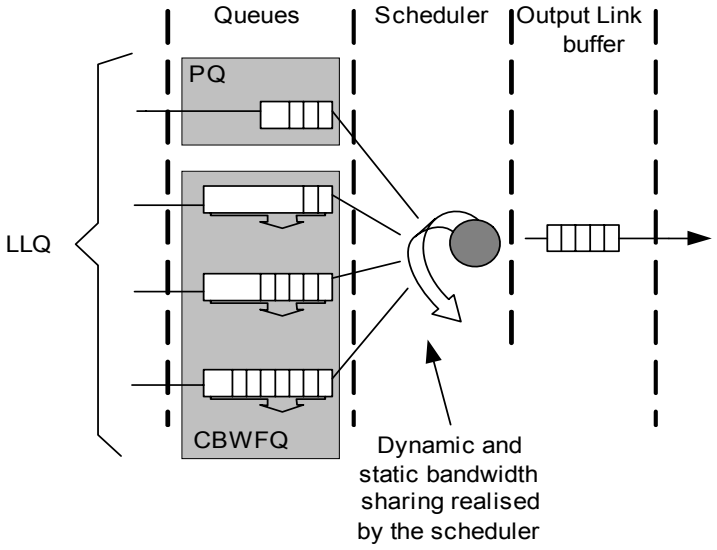


Figure 3.15 Bandwidth sharing

Bandwidth sharing can be static, such as CBWFQ, or dynamic, for example PQ. Or in the case of LLQ it is a combination of both. Bandwidth allocation is rather the predetermined (static) or varying (dynamic) bandwidth, which is reserved for each traffic class in order to provide sufficient QoS. The bandwidth allocated to each class is combined with a multiplexer to

determine the total bandwidth on the output link; which is the speed at which packets get transmitted from the network node. This is shown in figure 3.16.

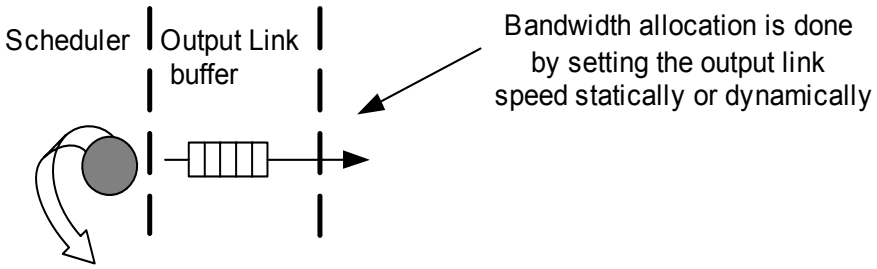


Figure 3.16 Bandwidth allocation

Both static and dynamic allocation has advantages and disadvantages, and these are a few of them compared in literature [35], [36] and [37]. Various proposals for new bandwidth allocation methods were made [38], [39] and [40]. Some of the most important conclusions are:

- Static bandwidth allocation is simple and avoids starvation, but wastes some bandwidth.
- Dynamic bandwidth allocation gives more bandwidth to the more demanding flows, but can result in starvation of slower or lower priority, flows.

Static provisioning is an old and not very popular technique, as opposed to dynamic provisioning. However, the question is how effective is dynamic provisioning really. The objective of the service provider is to sell as much as possible of the available bandwidth, while providing service guarantees. In terms of maximum utilization of the total link capacity, dynamic provisioning does not truly out-perform static provisioning.

Short interval utilization can be more effective with dynamic provisioning. In other words, the bandwidth that the customer is paying for, can track the demand for bandwidth much better. This seems to be a very convenient solution for the customer but the implications for the service provider needs to be considered. Due to the bursty nature of network traffic, the service provider would still have to loose a portion of the total link capacity in order to guarantee QoS at all times.

A further drawback of dynamic provisioning is the fact that it adds a lot of management traffic that normally receives high priority. In general, the loss of a certain portion of bandwidth, due to over provisioning in the static approach, is comparable to the extra load that mitigates the efficiency of the network in the dynamic approach.

It can thus be concluded that current methods of dynamic provisioning do not guarantee QoS with certainty during peek times. In order to make certain QoS guarantees static provisioning must be used. As described above, dynamic bandwidth sharing drastically improves utilization.

The combination of multimedia traffic with different quality requirements and statistical multiplexing makes it feasible to achieve good utilization as well as QoS with static provisioning. Consequently, the only significant drawback of static provisioning is thereby overcome. Figure 3.17 illustrates how bandwidth is allocated to the traffic classes by the use of class based static provisioning factors defined by Telkom SA Ltd. The higher priority classes naturally get higher provisioning factors. The most important conclusion drawn from the combination of LLQ and static over-provisioning is that is allows the service provider to statistically multiplex these classes together, which results in inter-class bandwidth sharing. This "overlapping" of the bandwidth is also illustrated in figure 3.17.

Current dimensioning, implying the choice of provisioning factors, is based on best practice recommendations. Intuition and educated guesses are widely used since no rigid equations exist for determining factors used in static over provisioning [19], [41]. All previous work that specifies mathematical equations for bandwidth allocation is either based on Poisson arrival processes or on analysis that is based on worst-case specifications. Poisson processes have proved to be insufficient to represent all network traffic [42] and are inadequate to perform worst-case analysis due to financial reasons and competition in the market.

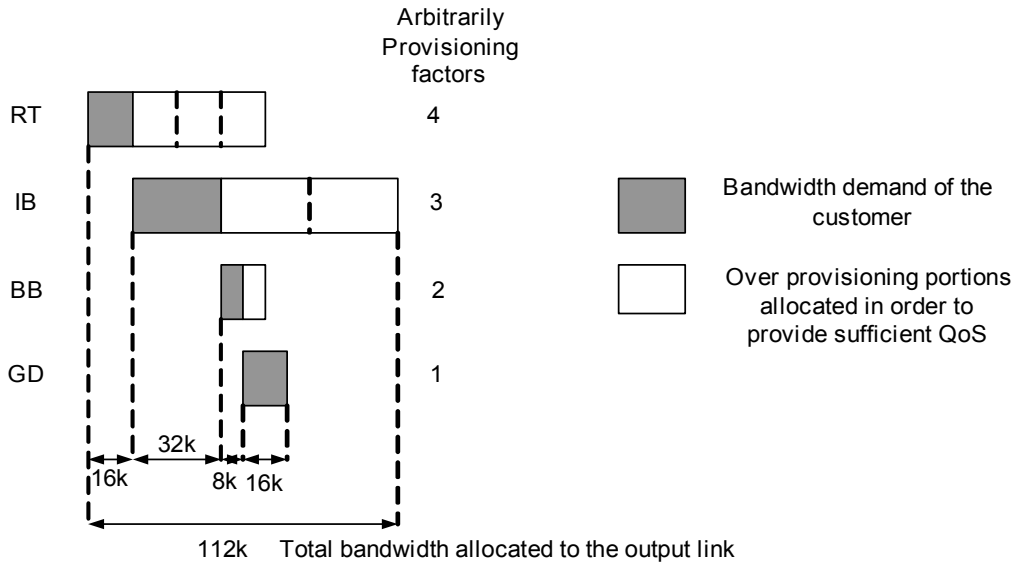


Figure 3.17 Bandwidth allocation and the concept of statistical multiplexing

This work uses accurate traffic modelling representative of the distinctive classes, along with the appropriate node architecture in simulations in order to generate QoS measurements as a function of provisioning values. This results in provisioning factors that are based on simulations rather than educated guesses. Chapter 4 summarizes a study of traffic modelling and motivates why certain traffic models are used for the simulation in Chapter 5 and 6.

3.8.2 Token Parameters

As described earlier in this chapter, the use of policers and shapers keep the arriving traffic in conformance to the traffic profile specified in the SLA. It is important that the token parameters are chosen such that they do not degrade the QoS. For instance, too many accepted bursts my result not only in the degradation of other classes, but also in unacceptable delays for the concerned class. Likewise, too small parameter values may result in too much loss. The determination of these values is additionally dependant on the output link speed and will inherently differ for various provisioning factors and package structures. Simulations to determine provisioning factors and token parameters will thus be performed simultaneously

3.9 CHAPTER SUMMARY

The layered structure of the OSI protocol stack provides a valuable layout of how different protocols work together and how network architectures are designed. It helps to isolate a problem and to define the problem boundaries. The functionality of the DiffServ architecture operates at the Network layer; hence additional overhead of lower layer protocols should be considered when dimensioning tasks are performed. DiffServ makes use of the IP header to distinguish between the traffic belonging to different classes. MPLS is also used in the DiffServ architecture to perform fast switching the core networks. Within the DiffServ domain various types of routers are defined, which are responsible for different functions. At the ingress node of a DiffServ domain, traffic gets separated in classes and a packet's behaviour at each node, throughout the domain, gets determined by the class it belongs to. The functions performed at the routers include classification, marking, policing, shaping, queuing and scheduling. The scheduler can be configured to perform static or dynamic bandwidth sharing.

Bandwidth allocation for the output link can also be performed statically or dynamically. Statistical multiplexing allows for an improved utilization property of static provisioning. Additionally, currently used values for provisioning factors leaves room for improvement, consequently static provisioning is considered. With the purpose of dimensioning bandwidth, it is identified that although analytical techniques may provide a good insight on the operation of the scheduling algorithm, they are only good to perform worst-case analysis and may not be adequate for characterizing the average performance. For this reason, chapter 4 describes methods to perform accurate traffic modelling to be used for simulation in chapter 5 and 6.

CHAPTER 4
TRAFFIC MODELLING

4.1 INTRODUCTION

Throughout the development of integrated high-speed networks over the last decade, traffic theory played a minor role in the design of it. Network provisioning was generally based on simple rules of thumb while considerable effort was, and still is being spent on the development of a variety of QoS mechanisms. In this context traffic theory implies the application of mathematical modelling to explain the *traffic- performance relation.* This relation links the traffic profiles, network capacity, resource utilization and the resulting performance.

- In the process of resource dimensioning *deterministic* or *statistical* services can be used. While *deterministic* services provide a very simple model to various applications, they tend to over-commit resources because they account for worst-case scenario [43]. However, *statistical* services significantly increase the efficiency of network usage by allowing increased statistical multiplexing of the underlying network resources. This comes at the expense of packets occasionally being dropped or excessively delayed. This chapter first investigates the characteristics of network traffic, and then shows how a distinction can be made between groups of traffic based on their statistical properties. Statistical descriptors mentioned in this chapter are described in Appendix A.

Various legacy statistical models are investigated to represent these different groups of traffic: some of the earlier traditional models as well as the latest in academic research. A summary of previous research [42], [44], [45], [46] is provided in this chapter to point out the advantages

and disadvantages; as well as specific properties that may be captured by the different traffic models. The most appropriate model for each group of traffic is chosen to simulate the traffic performance relation.

4.2 CHARACTERISTICS OF NETWORK TRAFFIC

As the applications that use public networks change, the statistical properties and nature of the traffic traversing the networks change as well. Especially if one distinguishes between the Plain Old Telephone System (POTS), data networks and the introduction of real-time (RT) services like voice and RT video to data networks. Over the last three decades various traffic models were derived empirically from traffic traces measured from public networks. In the past, these networks were mostly limited to circuit-switched telephone networks. As a result, most of the properties captured failed to characterize the traffic found on modern integrated networks.

Earlier models [47], [48], [49], [50] characterize network traffic in terms of well-known traditional statistical parameters such as the mean arrival rate, standard deviation, variance etc. along with a specific arrival distribution. More recently, network traffic of high-speed integrated networks proves to be much more complex than to be adequately characterized by these parameters. A traffic characteristic that is receiving much attention in recent literature is the *self-similarity* of the traffic.

4.2.1 Self-Similarity

Self-similarity can best be described as the phenomenon of traffic "looking" alike on different time scales. Intuitively, the critical characteristic of self-similar traffic is that there is no natural length of a "burst": at every time scale ranging from a few milliseconds to minutes and hours. Similar-looking traffic bursts are evident over a wide time range. It was also found that aggregating streams of such traffic typically intensifies the self-similarity ("burstiness") instead of smoothing it [51]. This is the most important difference between traditional statistical traffic models and more recent models, reflecting the self-similar phenomenon. Self-similarity can be described by visual interpretation as well as mathematical formulation.

Visual Interpretation

Figure 4.1 depicts an Ethernet traffic trace on 5 different time scales. This trace was measured on the Bellcore network in August'89 [51].

Starting with a time unit of 100 seconds (a), each subsequent plot is obtained from the previous one by increasing the time resolution by a factor of 10 and by concentrating on a randomly chosen subinterval, as indicated by the darker shade. The time unit corresponding to the finest time scale is 10 milliseconds (e). It is important to observe the following from figure 4.1:

- All plots look intuitively very "similar" to one another, in terms of their distribution and are distinctively different from white noise, implying an independent and identically distributed sequence of random variables.
- The scaling property (y-axis) and the absence of a natural length of a "burst": at every time scale ranging from milliseconds to minutes and hours, bursts consist of bursty sub periods separated by less bursty sub periods.

This scale invariant or "self-similar" feature of modern network traffic is drastically different from both conventional telephone traffic and traditional stochastic models considered in the literature. The latter, typically produce plots of packet counts that are indistinguishable from white noise after aggregating over a few hundred milliseconds. This pictorial "proof" of the self-similar nature of Ethernet packet traffic suggests that network traffic on one time scale is statistically identical, at least with respect to its second-order statistical properties, to network traffic on a different time scale. Thus, this motivates the use of self-similar stochastic processes for traffic modelling purposes.

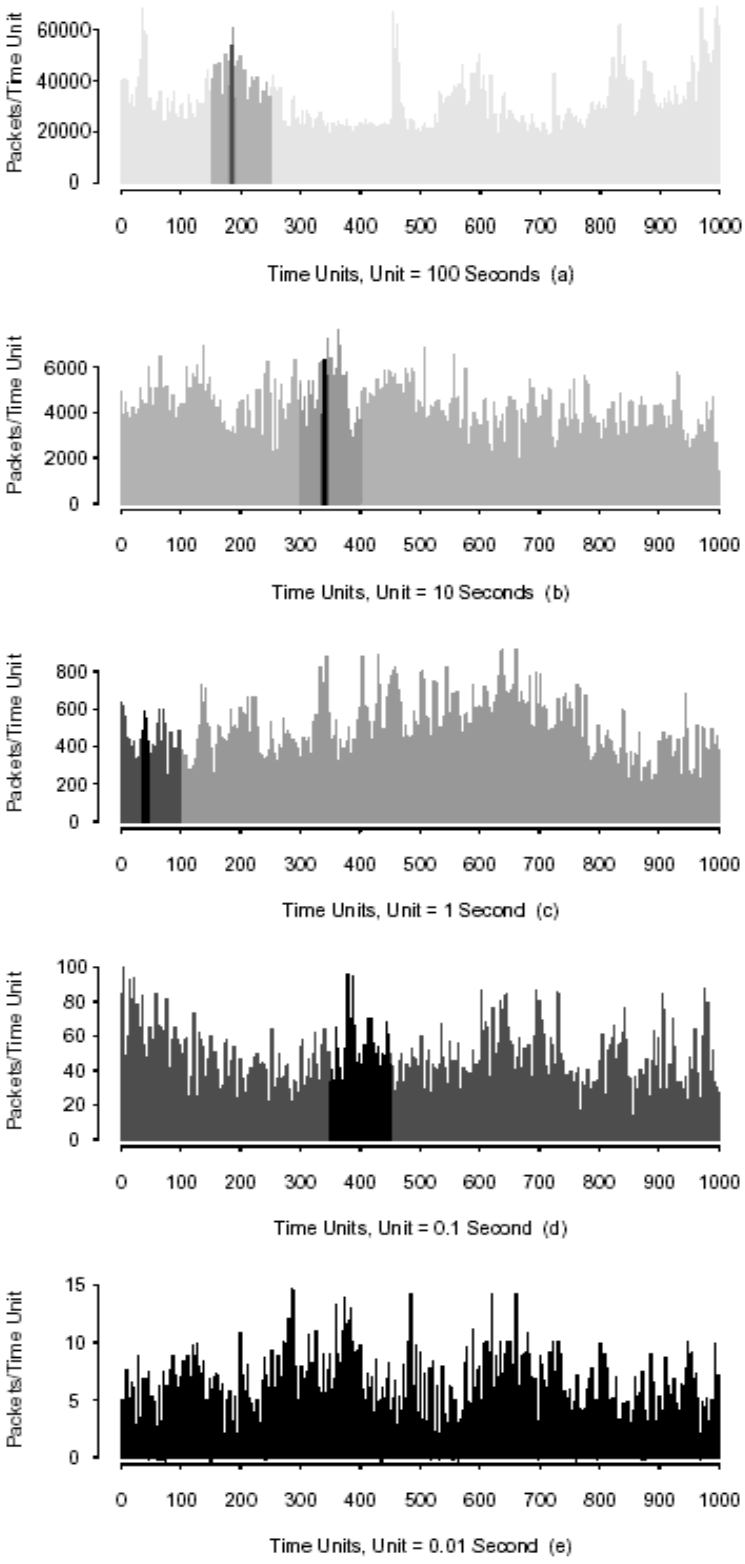


Figure 4.1 Graphical illustration of self-similarity. Each different grey tone represents a particular segment of traffic on different time scales.

Mathematical Formulation

The notion of self-similarity is not merely an intuitive description, but it can also be described mathematically. Over the last decade many researchers have tried to derive mathematical models to capture self-similar properties. Many have succeeded to define processes known as *exact,* or *precise, self-similar*. An exact self-similar process is a covariance stationary stochastic process with certain constraints, indistinguishable from its higher order moments. Real network traffic is not exactly self-similar. Although higher order characteristics differ, exact self-similar processes still prove to be much better approximations than earlier models. It is also the best way to describe self-similarity mathematically. In [44] it is described how a self-similar traffic model can be manipulated in order to get rid of the exact property.

The following analysis closely follows work done in [52], [52] and [53]. The exact self-similar concept is captured by the following mathematical definition:
Let $X = (X_t : t = 0, 1, 2, ...)$ be a *covariance stationary* (sometimes called *wide-sense stationary*) stochastic process; that is, a process with:

- Constant mean $\mu = E[X_t]$,
- Finite variance $\sigma^2 = E[(X_t - \mu)^2]$, and
- Autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)] / E[(X_t - \mu)^2]$, $(k = 0, 1, 2, ...)$ that depends only on k.

Particularly, X has an autocorrelation function of the form

$$r(k) \sim a_1 k^{-\beta}, \text{ as } k \to \infty, \tag{4.1}$$

where $0 < \beta < 1$, and $a_1, a_2, \ldots$ denote finite positive constants.

For each $m = 1, 2, 3, \ldots$, let $X^{(m)} = (X_k^{(m)} : k = 1, 2, 3, ...)$ denote a new time series obtained by averaging the original series X over non-overlapping blocks of size m. That is, for each $m = 1, 2, 3, \ldots$, $X^{(m)}$ is given by $X_k^{(m)} = 1/m\ (X_{km-m+1} + \ldots + X_{km})$, $(k \geq 1)$. Note that for each m,

the aggregated time series $X^{(m)}$ defines a covariance stationary process. The corresponding autocorrelation function is denoted by $r^{(m)}$.

The process X is called *exactly (second-order) self-similar* with self-similarity parameter $H = 1-\beta/2$ if the corresponding aggregated processes $X^{(m)}$ have the same correlation structure as X, i.e., $r^{(m)}(k) = r(k)$, for all $m = 1, 2, \ldots$ ($k = 1, 2, 3, \ldots$).

In other words, X is exactly self-similar if the aggregated processes $X^{(m)}$ are indistinguishable from X, at least with respect to their second order statistical properties. An example of an exactly self-similar process with self-similarity parameter H is *fractional Gaussian noise* (FGN) with parameter $1/2 < H < 1$, introduced by [54].

A covariance stationary process X is called *asymptotically (second-order) self-similar* with self-similarity parameter $H = 1-\beta/2$ if $r^{(m)}(k)$ agrees asymptotically (i.e., for large m and large k) with the correlation structure $r(k)$ of X given by (4.1). The *fractional autoregressive integrated moving-average processes* (FARIMA(p,d,q) processes) with $0 < d < 1/2$ are examples of asymptotically second-order self-similar processes with self-similarity parameter $d + 1/2$.

Intuitively, the most striking feature of (exactly or asymptotically) self-similar processes is that their aggregated processes $X^{(m)}$ possess a non-degenerate correlation structure as $m \to \infty$. This behaviour is precisely the intuition illustrated with the sequence of plots in Figure 4.1. If the original time series X represents the number of Ethernet packets per 10 milliseconds (plot (e)), then plots (a) to (d) depict segments of the aggregated time series $X_{(10000)}$, $X_{(1000)}$, $X_{(100)}$, and $X_{(10)}$, respectively. All of the plots look "similar", suggesting a nearly identical autocorrelation function for all of the aggregated processes.

Mathematically, self-similarity manifests itself in a number of equivalent ways:

- The variance of the sample mean decreases more slowly than the reciprocal of the sample size (slowly decaying variances), i.e. $\mathrm{var}\left(X^{(m)}\right) \approx a_2 m^{-\beta}$, as $m \to \infty$, with $0 < \beta < 1$. The

autocorrelations decay hyperbolically rather than exponentially fast, implying an uncountable autocorrelation function $\sum_k r(k) = \infty$ (long-range dependence), i.e., $r(k)$ satisfies relation (4.1).

- The spectral density $f(\lambda)$ obeys a power-law behaviour near the origin $\frac{1}{f-noise}$, i.e., $f(\lambda) \approx a_3 \lambda^{-\gamma}$, as $\lambda \to \infty$, with $0 < \gamma < 1$ and $\gamma = 1 - \beta$.

The existence of a non-degenerate correlation structure for the aggregated processes $X^{(m)}$ as $m \to \infty$ is in stark contrast to typical packet traffic models traditionally considered in the literature. Work in [44] proved that the exact self-similar models can be improved to resemble real network traffic more closely. A number of models along with their relevance regarding self similarity and the traffic characteristics will be evaluated in section 4.4 of this chapter.

4.2.2 Long and Short Range Dependency

To explain fundamentally how *Long Range Dependency* (LRD) differs from *Short Range Dependency* (SRD), let $\{X_t,\ t \in Z\}$ be a wide-sense stationary process, i.e. a process with stationary mean, μ, stationary and finite variance, γ_o, and stationary auto-covariance function, γ_k . The aggregate process can be obtained by averaging X_t over different non-overlapping blocks of size *m*, replacing each block by its mean as shown in equation (4.2).

$$X_t^{(m)} = \frac{X_{(t-1)m+1} + \ldots + X_{tm}}{m} \tag{4.2}$$

Let ρ_k^m denote the autocorrelation of $X_t^{(m)}$ at lag *k* and aggregate level *m*, and additionally the variance of $X_t^{(m)}$ as v^m. In [49] it is shown that the variance of traditional models decay asymptotically to m^{-1}, implying that

$$v^m \approx Km^{-1}, \tag{4.3}$$

where *K* is a positive constant. This resembles SRD.

Sample traces collected from network traffic showed that the variance decays slower than m^{-1}. LRD models comply with this slower decay by a variance decaying proportional to $m^{-\alpha}$ for $\alpha \in (0,1)$. In this case the autocorrelation will also be proportional to $m^{1-\alpha}$, i.e.

$$\sum_{k=1}^{m} \rho_k \approx Cm^{1-\alpha} \quad . \tag{4.4}$$

This leads to the defining property for LRD: It is impossible to sum the autocorrelation of a LRD process [49], implying that

$$\sum_{k=-N}^{N} \rho_k^m \to \infty \qquad (as\ N \to \infty). \tag{4.5}$$

Note that the definition of LRD is of an asymptotic nature. It does not describe the exact behaviour of the autocorrelation at any fixed finite lag. An important class of LRD processes is the *second-order self-similar process*, which has the property of preserving the second-order statistical properties over a range of scales. The variance of second order self-similar processes is given by [55]

$$v^m \approx Km^{2H-2}, \tag{4.6}$$

where K is positive constant and H is called the Hurst parameter which is related to α by $H = 1 - \alpha/2$. The Hurst parameter gives a measure of the self-similarity and long-range dependence of the traffic. From equation (4.6), it is evident that if $H=1$, the variance becomes constant, independent of the scale, while for $H = 0.5$, the variance is the same as that of short-range dependent models. LRD models need a Hurst parameter value that lies between 0.5 and 1.

In more general terms, a stochastic process is said to be LRD if specific activity at a certain time instance repeats over a relative large range of lags in the time plane. This dependency, or

correlation, is revealed by a process's autocorrelation function. Refer to Appendix A for mathematical detail on the autocorrelation function. LRD processes reveal a slow decaying autocorrelation, also known as a heavy-tailed distribution. Figure 4.2 shows an example of an autocorrelation structure belonging to a LRD process.

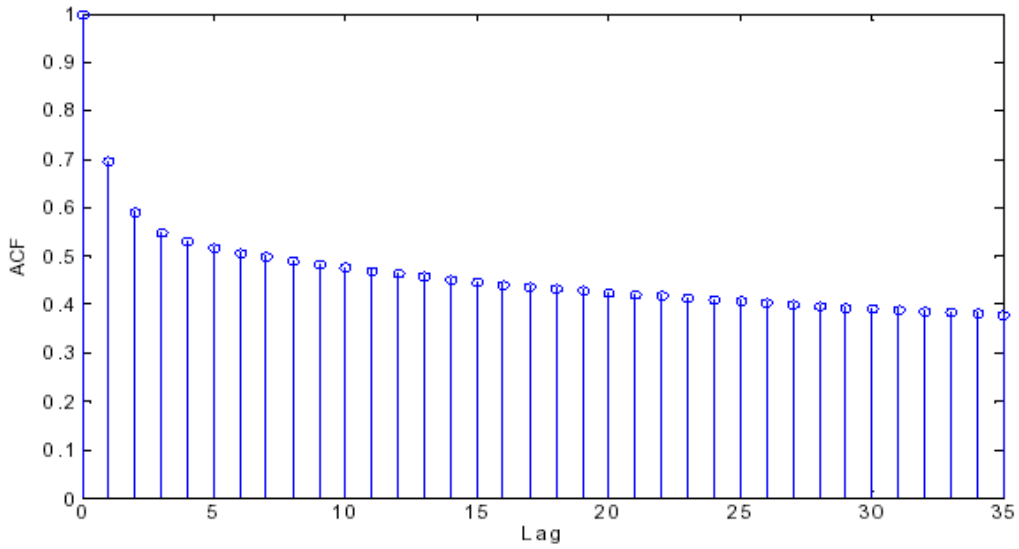


Figure 4.2 The slow decaying autocorrelation structure of a LRD process

A stochastic process is said to be SRD if a specific activity at a certain time instance repeats again at only relative small lags in the time plane. SRD processes have a correlation structure that is significant only for small lags. In other words, they have an exponential decaying correlation as illustrated in figure 4.3.

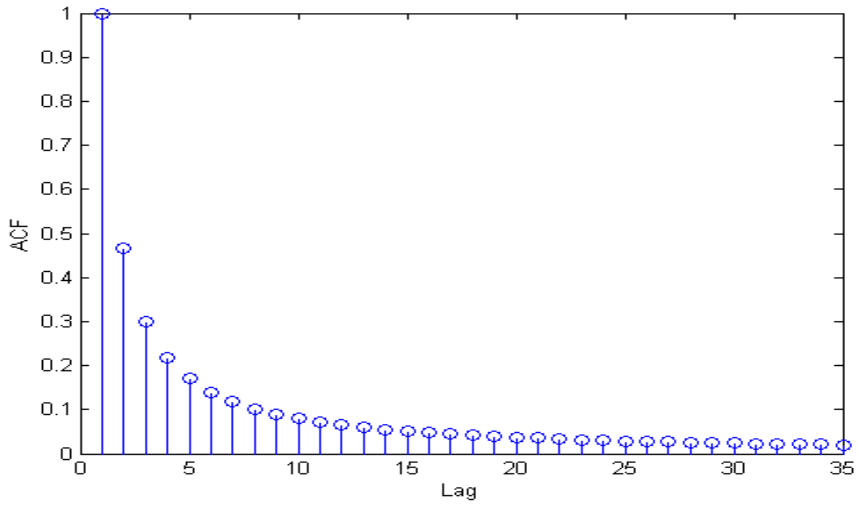


Figure 4.3 The fast decaying autocorrelation structure for SRD models

Modern day network traffic exhibits both LRD and SRD. It is thus important to choose appropriate traffic models capable of capturing these characteristics.

4.2.3 Burstiness

From a network point of view, specifically when looking at the trade-off between resource utilization and QoS, the burstiness of a traffic source is a crucial parameter. Burstiness directly affects queuing behaviour, particularly delay. It can be defined as the peak rate divided by the average rate over a measured period of time. As an example, two arrival processes with different parameters of burstiness are shown in figure 4.4. A largely bursty process (synthesized UDP traffic) is shown in (a) while (b) shows a less bursty process with an exponential distribution.

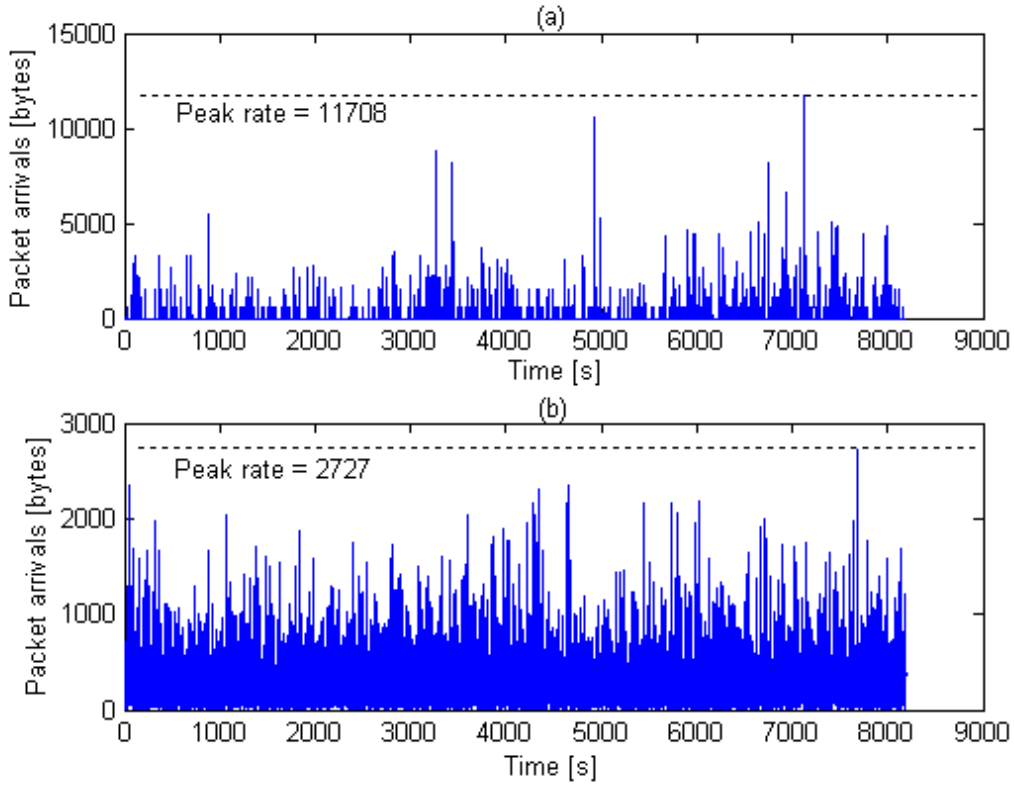


Figure 4.4 The burstiness of two arrival processes

Both the arrival processes in figure 4.4 have mean arrival rates of 300 bytes/s; however, there is a huge difference in their peak arrival rates as shown in the picture. Using the equation:

$$Burstiness = \frac{Peak \quad rate}{Average \quad rate}, \tag{4.7}$$

the burstiness can be determined. The first process, generated with the Multi-fractal Wavelet Model[6], has a burstiness of 39, while the second process, generated by the well-known exponential distribution, has a burstiness of 9. In this example, for instance tight bounds of zero loss- and delay rates have to be guaranteed. At least 11,708 Kbytes/s have to be allocated for the first trace compared to 2,727 Kbytes/s for the second. That is a difference of more than a factor of 4, for two processes with exactly the same average arrival rate. The tight bounds in

[6] The Multi-fractal Wavelet Model (MWM) is one of the traffic models to be evaluated later in this chapter.

this example are extreme, but the same principle applies with more realistic bounds. Rate limiters like policers and shapers, as discussed in chapter 3, have the ability to control burstiness to a certain extend.

This example illustrates that the burstiness of an arrival process is of great concern in the characterization of modern multimedia network traffic since it has a direct effect on the performance (QoS) of a network for specified bandwidth efficiency. The lower the burstiness of a traffic source, the better bandwidth efficiency can be obtained with a specified QoS. Unfortunately, modern network traffic exhibits high measures of burstiness. Figure 4.5 illustrates the balance between bandwidth efficiency, QoS and the burstiness of the arrival process.

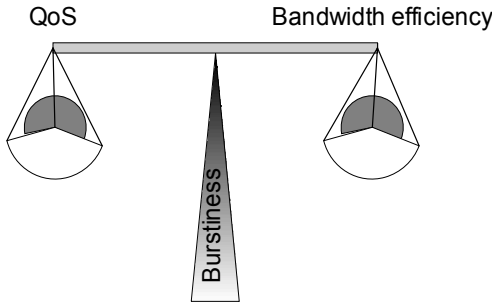


Figure 4.5 The balance between bandwidth efficiency, QoS and burstiness

From equation 4.7, it is known that the smallest quantitative value for burstiness is 1. This value could be obtained from an arrival process made up of periodic bursts of the same size. In this case, 100% bandwidth efficiency could be achieved when a server rate equal to the average arrival rate is used. This result in a minimum possible delay; i.e. only the serialization delay. Figure 4.6 depicts this situation.

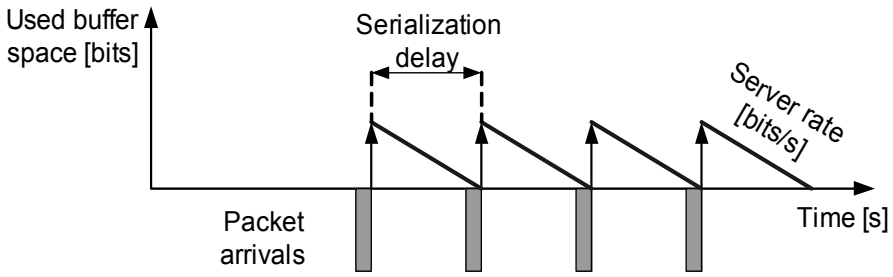


Figure 4.6 Minimum possible delay and maximum possible bandwidth efficiency

This situation is not realistic in terms of real arrival processes. For a given average data rate, packets arrive in bursts. Figure 4.7 shows the same number of packets arriving in the same total period of time and the same server rate applies. In this case, over the measured period of time, the bandwidth efficiency is still 100%, but more delay is introduced and more buffer-space is used because of a more bursty arrival process.

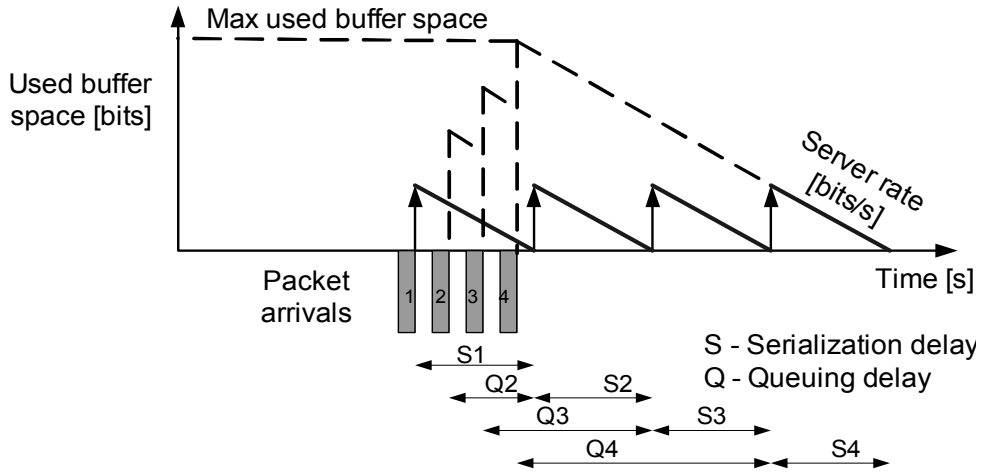


Figure 4.7 Burstiness introduces more delay and more buffer-space is used

From figure 4.7 it is obvious that the tighter the burst, the worse the queuing component of the delay. To compensate for this delay, the server rate can be increased. By doing this, the delay is improved at the cost of bandwidth efficiency. For practical arrival processes, the delay approaches infinity at 100% utilization. The more bursty a process gets, the worse its delay-utilization trade-off gets as depicted in figure 4.8.

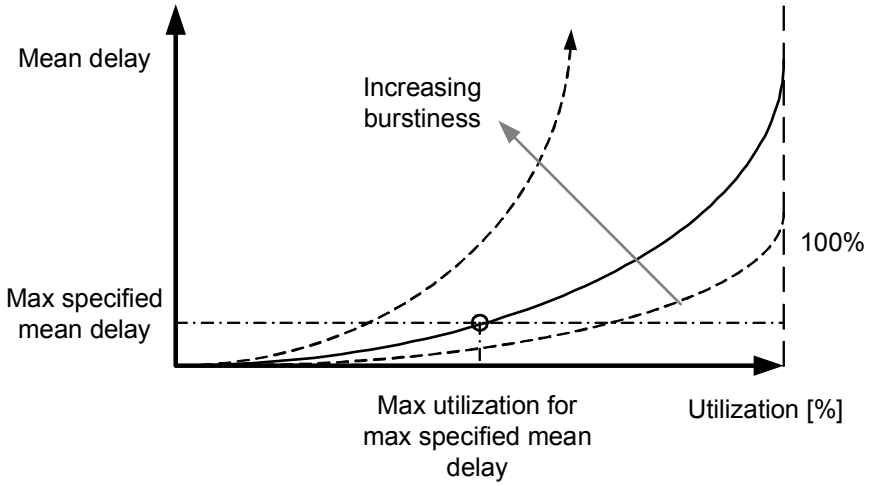


Figure 4.8 The trade-off between utilization and delay for bursty arrival processes

An additional way of dealing with this delay is by controlling the bursts in an arrival process. This can be achieved by using rate limiters like policers or shapers discussed in chapter 3. Rate limiters are mainly implemented for two reasons: Firstly, to prevent higher classes from depriving the lower classes of their allotted bandwidth, and secondly, to control burstiness on a link. To explain how rate limiters control burstiness, recall the functionality of the rate limiters and the functionality of the token bucket as discussed in chapter 3. Figure 4.9 illustrates how the rate limiter (in this case a policer), controls the burstiness of an arrival process at the edge of a network. The natural arrival process at the edge of the network is depicted in (a). The rate-limited traffic after it has been policed is shown in (b).

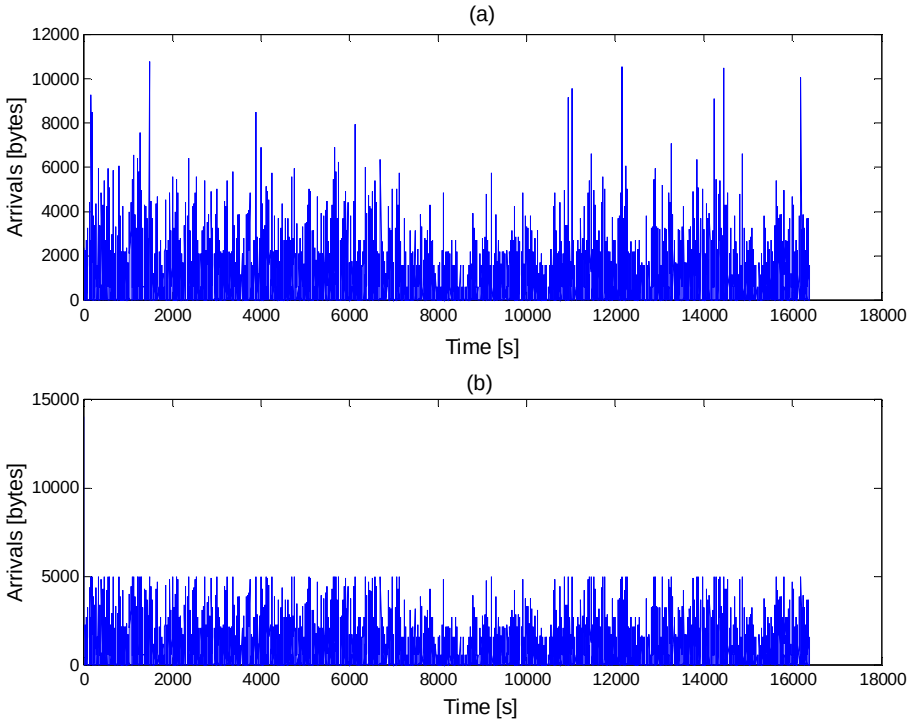


Figure 4.9 Bursty arrival processes can be controlled by rate limiters

In this case only the big bursts are eliminated. In general, the token bucket parameters discussed in chapter 3 can be chosen appropriately to control the traffic profile entering the network. Delay sensitive data is policed while delay tolerant data is shaped.

Packet-level traffic measurements from the Sprint IP backbone [56] have shown that traffic is burstier in a finer time granularity. Time granularity plays an important role in the accuracy of

simulations especially when packet level delay is analyzed. For example in the case of real-time voice where the maximum total mouth to ear delay budget is 200ms. To achieve an accuracy of 95%, time intervals of at least 10ms should be used. This becomes a tricky situation since real traffic traces as well as synthesized traffic of any source, especially UDP, are too bursty at fine sample periods (<30ms). Long off periods and relative large bursts result in low mean values and high burstiness. This figure for burstiness cannot be used to derive the token bucket parameters. Further, the mean value from such data is insufficient to use in bandwidth allocation equations: local high periods should be considered. Another factor that needs to be considered in dimensioning tasks is the protocol breakdown structure which in turn affects the distribution of packet sizes.

4.3 TRAFFIC DISTINCTIONS

4.3.1 Protocol Breakdown Structure

A study [57] of the protocol breakdown structure shows that TCP is by far the dominating transport layer protocol. UDP took second place with less than 10% of the amount of TCP traffic. The Encapsulating Security Protocol (ESP) was third with just more than 1% of the total amount of TCP traffic. The rest of the transport layer protocols become insignificant when in the analysis of packet size distribution (PSD) for modelling purposes. Figure 4.10 shows a histogram of the top 18 protocols observed on the Sprint backbone network on the 7th of April 2003 [57]. Various other traces have been analysed that show similar results.

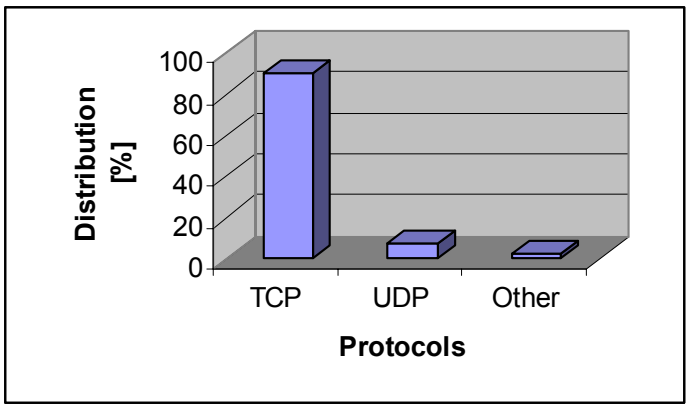


Figure 4.10 Transport Layer Protocol breakdown structure

This breakdown structure shows that it is only necessary to analyse TCP and UDP traffic to get an adequate PSD for simulation purposes. Analysis of various traces over the last four years has proven that this dominance of TCP and (to a lesser extent) UDP is not changing drastically. Many researchers expect UDP traffic to increase dramatically relative to TCP. However, one has to keep in mind that not only the demand for real-time applications (mostly using UDP) is growing fast, but also the demand for non-real-time applications (mostly using TCP). The ratio: TCP vs. UDP traffic loads over the last four years is shown in figure 4.11. This data is a result of the Sprint IPMON DMS network analysis [57].

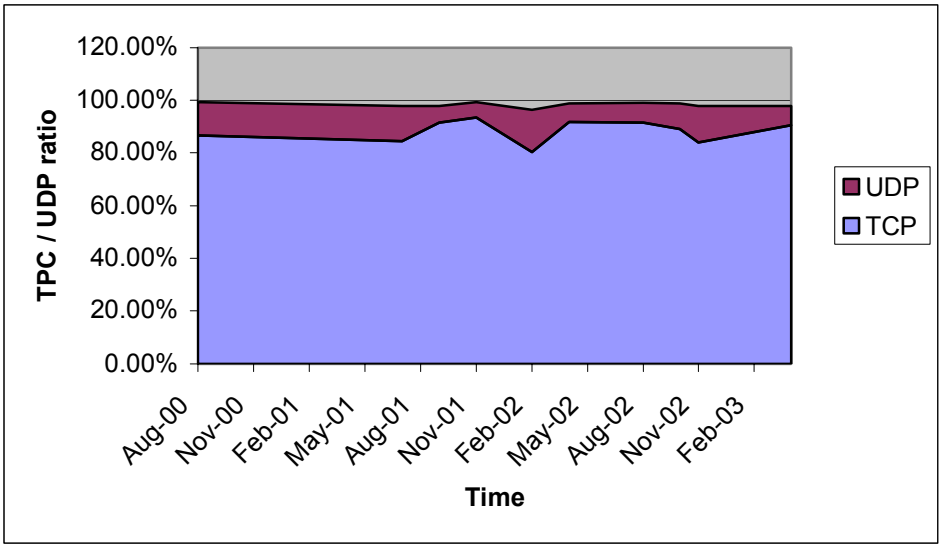


Figure 4.11 TCP and UDP traffic on the Sprint backbone over the last four years

From figure 4.11 it is clear that there is no drastic change or trend present in the protocol breakdown structure. Hence, the distribution as shown in figure 4.10 is assumed and utilized.

4.3.2 Packet Size Distribution

Packet size distribution (PSD) is a characteristic of network traffic that describes the size of packets travelling across the network. If each packet in production workloads had the same size, there wouldn't be any need for a distribution statistic. But in reality, production networks are host to a wide range of packet sizes, and the packet size distributions of most production workloads are far from ideal.

Network analyzers often generalize by saying that about half of the packets are very large and the other half are very small, but packet size distribution plays a significant role in the performance of networks. It determines the efficiency of the workload, which in turn, has a direct impact on how efficiently the network bandwidth is utilized. An even more important consideration with regards to the PSD is the allocation of token bucket parameters as described in chapter 3. It is thus a concern to investigate it and formulate more accurate distributions.

The applications, using a specific traffic class, have approximately the same performance goals, and thus use the same transport layer protocols. These protocols each have a profile describing the distribution of their packet sizes. Research in [56], [57] and [58] has been done to characterize the PSD of protocols distinctively. Figure 4.12 depicts typical PSD for UDP and TCP respectively.

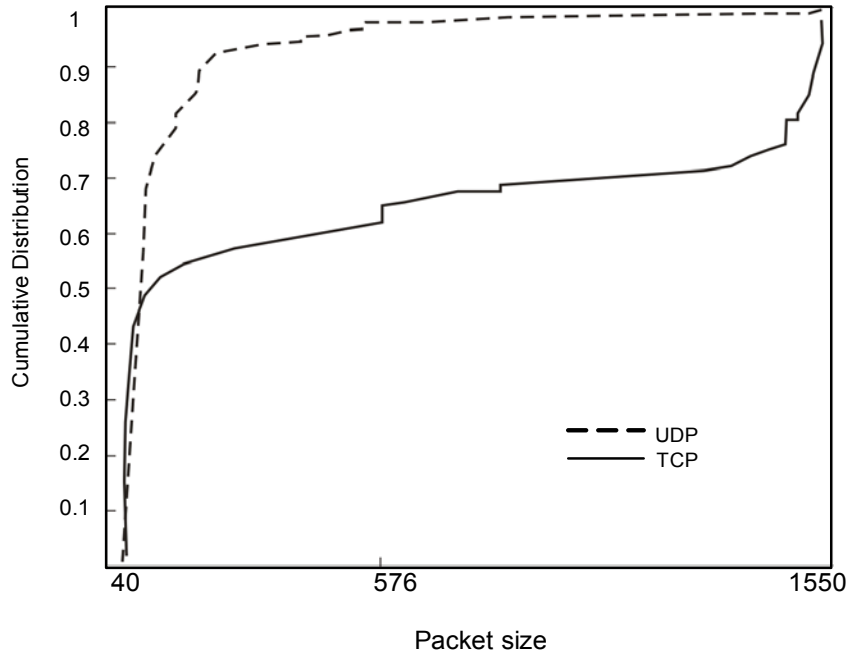


Figure 4.12 Packet size distributions for UDP and TCP

From figure 4.12 it is clear that average UDP packet sizes are smaller than those of TCP and that more than 70% of all UDP packets are between 40 and 80 bytes in size. Nearly 50% of the TCP packets lie in the vicinity of 40 bytes. A large portion of these bytes are TCP

acknowledgement (ACK) packets. In general most of the data-carrying TCP packets are significantly larger in size. The smaller a TCP packet, the less efficient it is because of overheads. A more detailed study on this topic appears in [59].

In order to dimension token bucket parameters, specifically the maximum permitted bursts, packet sizes have to be considered. Current best practice recommendations used by vendors specify to use the maximum transmission unit (MTU). This again results in a deterministic result which is definitely not the best result with regards to utilization. This dissertation investigates the packet sizes for UDP and TCP respectively. For simulation purposes the PSDs depicted in figure 4.12 are used. As described in chapter 3, traffic entering a DiffServ node is divided into different classes. It is needed to investigate what the PSD look like for the various traffic classes.

4.3.3 Class Based Traffic Distinction

A distinction is made according to the transport layer protocols. Most applications belonging to the RT class make use of the UDP or an equivalent real-time protocol in order to achieve real-time objectives. All the other classes serve applications requiring acknowledgement of data that is received in order to reassemble fragmented data streams. Consequently, most of these applications make use of TCP for ordered connection-oriented delivery.

UDP characteristics will be used for the RT class, although only small packet sizes (below 200 bytes) will be used in simulations since RT applications only generate small packet sizes. TCP characteristics will be used for the rest of the classes. Analysis of separate UDP and TCP traces revealed that UDP exhibits SRD while TCP exhibits LRD. Hence, a SRD and a LRD traffic model are needed for UDP and TCP respectively.

Results of [44] prove that traces generated with the Multi-fractal Wavelet Model (MWM) give the best correlation with wide area TCP traffic. The same model was used for UDP, but the MWM did not give the same good result for UDP. Another model is required for the more bursty UDP traffic. In the following, a number of traffic models are discussed and compared

with more emphases and detail provided for the Markov Modulated Poisson Process (MMPP) and the MWM to be used for UDP and TCP respectively.

4.4 TRAFFIC MODELS

Traffic models play a significant role in the characterization and analysis of real network traffic. These models are used in the planning and dimensioning of network resources to enhance network performance and resource utilization. As we strive towards providing NGN services, the emphasis of research and development shifts to high-speed networks providing integrated services at certain QoS specifications. Consequently, the role of accurate traffic models in network design and network simulation becomes ever more crucial. Accurate traffic modelling enhances the understanding of complex networks and the behaviour of real networks under these complex conditions. It allows studies on the effect of different model parameters and network configurations on the network performance through simulation.

Modern data networks host a variety of traffic types and a large number of traffic sources, and are still expanding at a growing rate. The result is a complex traffic system from a statistical point of view, with accurate traffic modelling being a difficult task. Throughout decades of research in the field of network traffic modelling, few models were able to capture the majority of the concerned real traffic characteristics [42], [44], [51]. Another factor that adds complexity for performing accurate traffic modelling is the length of traffic traces for use in analysis: The length of a traffic trace has a direct influence on the statistical confidence interval [60].

The quality of a traffic model is determined by two factors: the *model performance* and the *computational efficiency* [44]. The former deals with the ability of the model to capture various statistical properties of network traffic. The latter addresses the complexity of the model and the computational complexity needed to fit such a model to an actual traffic trace and to synthesize a large traffic trace. Most traditional models such as Poisson, Markov and Auto Regressive Moving Average (ARMA) are simple from a computational point of view, but their performance in representing network traffic is not adequate for all network traffic

[42]. Later models (that exhibit the concept of self-similarity) give much better accuracy, but the model parameters are difficult to derive and very little academic literature on the subject exists. Despite the computational complexity, the self-similar models are considered in current research because of the inadequate performance of traditional models.

The discovery of LRD in modern network traffic such as Ethernet data [51] and variable bit rate (VBR) video traffic [61], lead to the development of self-similar traffic models. Traditional models are unable to capture the LRD of some traffic sources, which results in an over-optimistic model as LRD can lead to higher packet loss than predicted by these models. To add to the problem, a further statistical property, the co-existence of LRD and SRD, has been found in VBR video traces [61].

Accurate traffic modelling is thus a challenging task. The derivation of yet another new self-similar traffic model is beyond the scope of this thesis. Comparison and evaluation of various models is done and the most applicable models are used to perform resource dimensioning tasks.

4.4.1 Short-Range Dependent Traffic Models

The characteristics of LRD and SRD described in section 2 of this chapter can be captured by using the appropriate traffic model. SRD traffic models produce traffic traces with a correlation structure that decays exponentially. These types of models were used initially due to their computational simplicity and the tractability of their results. A number of SRD traffic models are summarized in the following.

Autoregressive Moving Average (ARMA) Models

An ARMA(p,q) process is one of the traditional methods for modelling network traffic as a time series. ARMA(p,q) is a combination p^{th} order Autoregressive process, AR(p), and a q^{th} order Moving Average Process, MA(q). The stationary time series is of the form

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_1 - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \tag{4.8}$$

Autoregressive part Moving average part

or

$$\phi_p(B) X_t = \theta_q(B) \varepsilon_t \tag{4.9}$$

where $B^j X_t = X_{t-j}$, p is the order of the autoregressive part, q is the order of the moving average part, ϕ_i is the i^{th} autoregressive parameter, θ_j is the j^{th} moving average parameter and ε_t is a time-dependent exciting term. Here $\phi_p(B)$ and $\theta_q(B)$ are polynomials in B of degree p and q respectively.

The ARMA(p,q) model is used for processes where the current process value depends on both the previous process values and previous excitation terms. It can only model a stationary time series due to the time-independence of the coefficients of the model. If the time series to be modeled shows growth or any time-dependence, the ARMA-modelling technique is not suitable.

ARMA models were used to model earlier telephone traffic, end-to-end delay and loss of Internet traffic, and variable bit-rate video traffic [62] and [63]. But because the traffic is usually non-stationary, these models do not provide satisfactory results.

Autoregressive Integrated Moving Average (ARIMA) Models

The Autoregressive Integrated Moving Average (ARIMA) was developed with the intention to overcome the stationary limitation of the ARMA model. It uses lags and shifts in the historical data to uncover patterns (e.g. moving averages, seasonality) and predict the future. It models a non-stationary time series, which exhibits a homogeneous variation around a local trend. The series will become stationary if the local trends are removed. ARIMA is a method for

determining how much of the past should be used to predict the next observation (length of weights) and the values of the weights. The ARIMA (p,q,d) model is described by

$$\phi_p(B)W_t = \theta_q(B)\varepsilon_t \tag{4.10}$$

where

$$W_t = \nabla^d X_t, \tag{4.11}$$

and

$$\nabla^d = (1-B)^d \tag{4.12}$$

$\phi_p(B)$ and $\theta_q(B)$ are defined as for the ARMA process. ARIMA models were previously used for network traffic modelling, but nowadays they are used in economic financial models and business intelligence software. The seasonal ARIMA (SARIMA) model was proposed in [64] to accommodate seasonal behaviour.

Markov chains and Markov Modulated Processes

A Markov chain is a stochastic process where the future values only depend on current values. Markov chains can be used to model a source that has a finite number of states and the probability of the next state only depends on the current state of the source. The activity of a traffic source can usually be described by a finite amount of states. Increasing the amount of states will result in a more accurate model, but this will increase the processing power needed to compute the process.

The state diagram in figure 4.13 shows a two state Markov chain with the two states, 1 and 2, and their transition probabilities, α and β. The transition probabilities are the probabilities of moving from one state to another. The probabilities to remain in the current state is 1-α and 1-β, respectively.

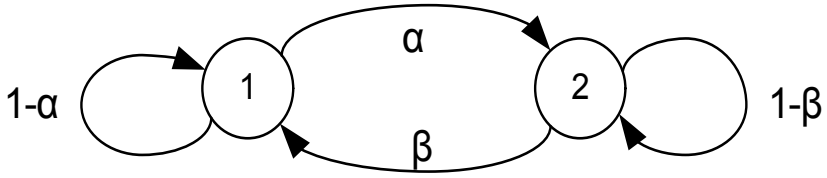


Figure 4.13 State diagram of a Markov chain

Markov chains are usually combined with other statistical processes to obtain more accurate results. Here, the current state of the Markov process controls (modulates) the probability law of the traffic generating process. These models are also referred to as double stochastic processes. One of the most popular modulated processes used for network traffic modelling is the *Markov modulated Poisson process* (MMPP). This modulated process combines a Markov process with a Poisson process. Each state of the Markov process represents a different probability distribution for the Poisson process. When the state of the Markov chain changes, so does the rate of the Poisson process.

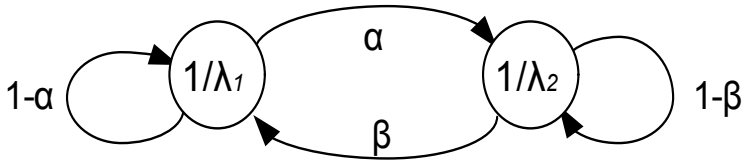


Figure 4.14 State diagram of a Markov modulated Poisson process

Figure 4.14 shows a state diagram of a 2 state MMPP. Observe that while in state the k, packet arrivals occur according to a Poisson process with rate λ_k. This allows one to model, for example, a single time-varying traffic source, such as continuous voice or data traffic, or a combination of traffic sources such as the combined arrival rate of packets for a mixture of continuous voice and data sources [65], [66].

Related to the MMPP is the *Markov modulated Bernoulli process* (MMBP). The MMBP uses a Markov process to modulate a Bernoulli process instead of a Poisson process. The Bernoulli process is the discrete equivalent of a continuous Poisson process. It does not produce a continuous output, but a discrete output. MMBP models are used to model a mixture of discrete voice and data traffic [65].

Another Markov modulated process is the *Markov modulated fluid model* (MMFM). A fluid model characterizes the network traffic as a continuous stream with a changing flow rate. When this is combined with a Markov process, each state of the Markov process represents a different flow rate for the fluid model, much like the MMPP. While in state s_k the traffic arrives according to a fluid model with rate λ_k. The MMFM is used to model large volume traffic traces where the individual packets have little or no effect on the performance of the network, and the focus is on the aggregate flow of traffic. MMFM was used to model variable bit-rate video sources [67].

Transform-Expand-Sample Models

Transform-Expand-Sample (TES) models are non-linear regression models with the aim to capture the autocorrelation structure (Appendix A) and marginal distribution of the empirical data. It can generate a variety of SRD autocorrelation functions (ACF), to approximate an empirical sample, while guaranteeing a matching marginal distribution to that of the empirical data. Refer to [44] for a detailed description of how traffic can be generated with TES models. TES was used to model compressed video streams such as MPEG [68], [69] because they give rise to highly correlated traffic streams.

4.4.2 Long-Range Dependant Traffic Models

Traditional traffic models such as those described in Section 4.4.1 have an exponentially decaying ACF. In [70] it is shown that the autocorrelation of actual Ethernet traffic decays to zero at a much slower rate than exponential. Thus strong correlations are present in actual network traffic. These strong correlations are a result of the network traffic's bursty nature. A number of LRD traffic models are summarized by the following models.

Fractional ARIMA (FARIMA)

FARIMA is a LRD model that is an extension of the ARIMA model of the form:

$$\alpha(B)\nabla^d X_t = \beta(B) z_t \tag{4.13}$$

where $\alpha(B)$, $\beta(B)$ and ∇^d are defined as in section 4.4.1 but in the case of FARIMA the parameter *d* only assumes values $0< d \leq ½$. *The value of d* determines the rate of decay of the autocorrelation. The relationship between *d* and the Hurst parameter is given by $d = H - 0.5$. The closer the value of d gets to ½, the stronger the long-range dependence of the process will be.

The calculation of the FARIMA model's parameters is complicated and therefore a disadvantage. Additionally, the generation of a synthetic trace is also computation intensive. Refer to [44], [71] for a detailed description of the FARIMA model.

Fractional Brownian Motion Model

The normal Brownian motion, denoted by $\{B_t\}$, has the following properties:

- For any t ≥ 0 and t0 ≥ 0 the increment $B_{t+t_0} - B_t$ is normally distributed with mean 0 and variance σ^2t.
- If $0 \leq t_1 \leq t_2 \leq ... \leq t_n$, then the increments $B_{t_2} - B_{t_1}, B_{t_3} - B_{t_2}, ..., B_{t_n} - B_{t_{n-1}}$ are independent variables.
- $B_0 = 0$ and B_t are continuous functions of *t*.

The fractional Brownian motion, denoted by $\{fB_t\}$, differs in the sense that $B_{t_2} - B_{t_1}, B_{t_3} - B_{t_2}, ..., B_{t_n} - B_{t_{n-1}}$ is no longer independent and has variance $\sigma^2 t^{2H}$, where *H* is the Hurst parameter. The process displays long-range dependence for values of $0.5 < H < 1$. For a complete description of modelling traffic with the fractional Brownian motion, refer to [46] and [72].

Fractional Brownian motion is a popular approach to model WAN, LAN, and Internet traffic [46], [72] and [73]. Unfortunately the Fractional Brownian motion has limitations when

modelling network traffic which has non-Gaussian distributions, particularly its failure to capture the combination of SRD and LRD correlations.

Wavelet Models

Wavelet models are used to describe a process in terms of a combination of waves that differ in size and wavelength (or frequency). It can be compared to the Fourier transform, but unlike the Fourier transform, a wavelet has finite energy and time extent, allowing the successful application to the analysis of transient, non-stationary, and time-varying phenomena in general.

Wavelet models are not exactly long-range dependent, but they are able to capture the statistics of real traffic traces, especially the second-order statistics that define long range dependency. One of the major advantages of wavelet analysis is that it is not computationally intensive.

Multi-resolution wavelet analysis makes use of two closely related basic functions, namely the mother wavelet $\psi(t)$, and the basic scaling function $\phi(t)$. A signal can be represented by the mother wavelet and a basic scaling function, using the series expansion

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \psi_{j,k}(t) \tag{4.14}$$

The variables j and k are integers called the dilation or scale variable, and the shifting or translation variable respectively. The real numbers c_k are the scaling coefficients and $d_{j,k}$ are the wavelet coefficients, also called the (partial) discrete wavelet transform. The goal with the wavelet expansion, like the Fourier expansion, is that the coefficients provide more useful information about the signal than is directly noticeable from the signal itself.

A general discussion of the scaling functions and wavelet functions is beyond the scope of this text. For more detail on these functions, refer to [44] and [74]. Results of [44] conclude that

the Multi-fractal Wavelet Model (MWM), using the Haar wavelet with some constraint based manipulations, give the best performance[7] of all the investigated traffic models. The scaling- and wavelet functions of the Haar wavelet are thus described in the following text, along with constraints that have to be enforced on the synthesized traffic.

The Haar Wavelet

The Haar scaling function and wavelet function depicted in figure 4.15 correspond to:

$$\phi(t) = \begin{cases} 1, & if \quad 0 \le t < 1 \\ 0, & otherwise \end{cases} \tag{4.15}$$

and

$$\psi(t) = \begin{cases} 1, & if \quad 0 \le t < 1/2 \\ -1, & if \quad 1/2 \le t < 1 \\ 0, & otherwise \end{cases} \tag{4.16}$$

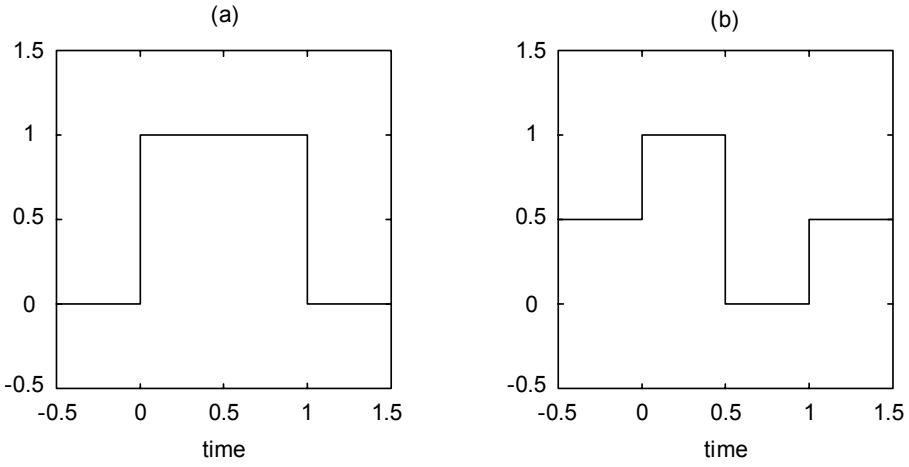


Figure 4.15 The Haar scaling function (a) and the Haar wavelet function (b)

The coefficients for the Haar scaling and wavelet functions are calculated recursively as follows [45]:

[7] This performance evaluation is based on TCP traffic. Other models are still considered for UDP.

$$c_{j,k} = \frac{c_{j+1,2k} + c_{j+1,2k+1}}{\sqrt{2}} \tag{4.17}$$

and

$$d_{j,k} = \frac{c_{j+1,2k} - c_{j+1,2k+1}}{\sqrt{2}} \tag{4.18}$$

Observe that the scaling coefficient is the scaled mean of two consecutive samples, while the wavelet coefficient is the scaled difference of consecutive samples. The following diagram illustrates how the coefficients are determined recursively.

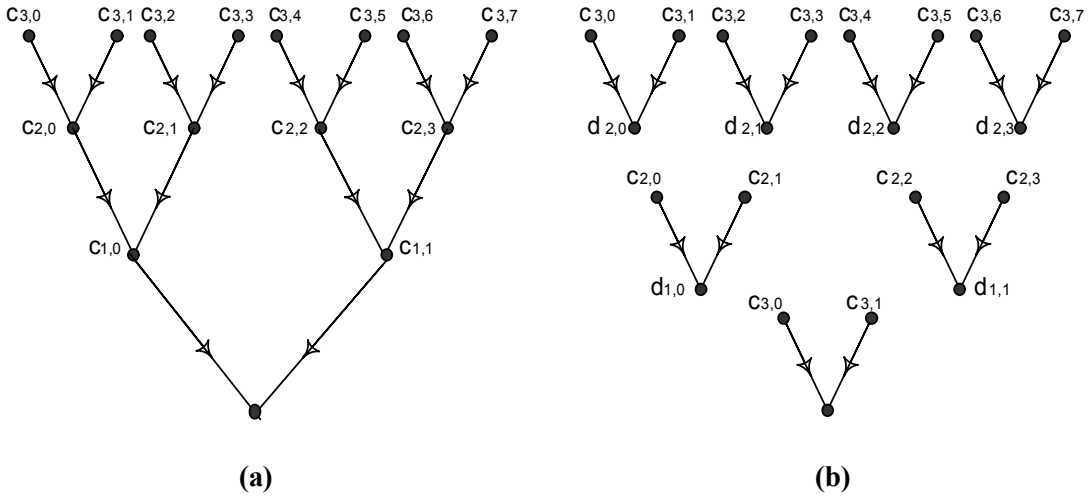


Figure 4.16 Binary tree of (a) scaling- and (b) wavelet coefficients from fine to coarse scale

Figure 4.16 shows 3 levels of coefficients. Accuracy of the traffic approximation improves with an increase of levels included in analysis and synthesis. According to [45], the equations for inverse wavelet transform (or synthesis) can be obtained by rearranging equations (4.17) and (4.18) to

$$c_{j+1,2k} = \frac{c_{j,k} + d_{j,k}}{\sqrt{2}} \tag{4.19}$$

and

$$c_{j+1,2k+1} = \frac{c_{j,k} - d_{j,k}}{\sqrt{2}} \tag{4.20}$$

A step-by-step fitting and synthesis procedure is provided in chapter 5. For further explanations and examples on the scaling and wavelet functions and coefficients refer to [44], [45] and [74].

4.4.3 Discussion

The choice of traffic model for optimization purposes is crucial since dimensioning values derived from simulations are a direct result of the node architecture, and in this context more importantly of the arrival process. The bursty arrivals of modern day network traffic cause extensive buffer overflows that are not predicted by traditional traffic models. Traditional models lead to over-optimistic queuing prediction, and as stated in chapter 2, queuing delay is the main controllable form of delay in a network.

Extensive work performed in [44], demonstrates that the MWM is currently most suitable to model Ethernet traffic. An important observation however is that more that 90% of this traffic is TCP and the same conclusions can thus not be assumed for UDP. The same experiments presented in [44] were performed for UDP traffic. Most of the characteristic descriptors revealed a similar correlation between the real traffic trace and the synthesized trace. However, one aspect that proved to be inadequate was the burstiness of the synthesized trace. The MWM tends to overemphasize the burstiness. It was argued that this was due to the model's tendency to capture the mean as well as long off-periods present in real UDP traces. Experiments resulted in burstiness differing by almost a factor 2.

The long off-period characteristic of the UDP traces leads to experiments with ON-OFF traffic models, specifically the MMPP. Experiments show good correspondence for the marginal distribution, ON-OFF probabilities, mean rate, peak rate, burstiness and queuing behaviour. The autocorrelation for different time scales did not prove to be good. However, supported by [70], [75], [76], [77] and [78] it is argued that for mission-critical traffic, only limited traffic

time-scales are of concern. Of greater importance for this kind of traffic is the burstiness that directly affects the queuing behaviour. The MMPP is thus used to synthesize UDP traffic for the RT class in simulation.

4.5 CHAPTER SUMMARY

This chapter provides background on network traffic characteristics and modelling. The concept of *self-similarity* is explained by means of visual interpretation and mathematical manifestations. The evidence of *long range dependency* (LRD) and *short range dependency* (SRD) in network traffic was introduced and explained. The chaotic way in which network resources are demanded, results in the *bursty* nature of network traffic. Methods to measure and quantify these characteristics are explained.

Network traffic can be distinguished not only by these statistical characteristics, but also by means of protocols. Packet size distribution and arrival processes differ among protocols, particularly the transport layer protocols (TCP and UDP). These protocols also happen to be the distinguishers of mission-critical applications and more tolerant applications.

A number of SRD and LRD traffic models are briefly described with reference to more detailed description in other literature. Previous research [44] proves that the MWM is most applicable for modelling TCP traffic. Our own experiments, along with previous findings, conclude that the MMPP is more applicable for UDP traffic in the concerned simulation environment. In chapter 5 The MWM and the MMPP are used to simulate TCP and UDP arrival processes respectively.

CHAPTER 5
SIMULATION SETUP

5.1 INTRODUCTION

In the process of resource dimensioning in a mixed traffic environment, various aspects have to be considered. These aspects are thus included in simulation and involve the compliance of QoS parameters described in chapter 2. Traffic flows should not suffer from poor bandwidth due to misbehaving flows. Furthermore, bandwidth is limited and hence should not be wasted. To include these aspects, the most appropriate architecture and methods for bandwidth allocation and sharing, as described in chapter 3, are used in analysis. Simulations have to be representative of real network situations, especially the network traffic models covered in chapter 4. Further, the dimensioning algorithm should be feasible from an implementation point of view and should scale well.

Active queue memory management performs selection of packets to drop during congestion (when resources are too little for the short term traffic profile). However these mechanisms are particularly not considered in this work since the goal is to provide adequately to avoid deliberate packet dropping.

This chapter describes the simulations performed to refine the values used for resource dimensioning. It first describes how this work relates to previous research and how this research extends the previous work in section 2. Real traffic traces first have to be analyzed in order to model (or synthesize) a corresponding trace. Some aspects regarding the traffic trace to be used are mentioned in section 3. Section 4 discusses theoretical considerations with

reference to previous chapters. The simulation environment and test bed are described in section 5. Lastly, the specific QoS parameters to comply with are quantified.

5.2 RELATED WORK

With regards to resource dimensioning, the Decomposition method proposed by [29] is a fundamental approach to partitioning the network into isolated servers, and bases the end-to-end delay analysis on the local delay analysis of the isolated servers. This dissertation extends the work of [29] by isolating the server and additionally decomposing the various contributors to the total delay, to analyze them separately. It also considers loss analysis.

A node architecture based on the Virtual Time Reference System (VTRS) is proposed in [79]. The intension of this architecture is to provide scalable support for multiple QoS guarantees. Four traffic classes are defined: Guaranteed service, Premium service, Assured service, and Best Effort service. Weighted Fair Queuing (WFQ) was proposed for the scheduling algorithm. The same node architecture is used in this study, but the scheduling discipline is extended to LLQ.

The work of [80] presents the partitioning of bandwidth among competing applications share a single link. Our work considers this partitioning among predefined traffic classes.

The authors of [81] used a fuzzy logic-based description to obtain a user-oriented ordering of QoS parameters. With the architecture considered in this study, users are supplied with the guarantees of each class. It is then up to the users to decide how much bandwidth they need in each class.

To substantiate this analysis, it is assumed that the traffic modelling to be used is reasonably accurate. This assumption can be made based on previous research, as presented in [44] Based on traffic characteristics studied in chapter 4, it is known that self-similarity and LRD are ubiquitous properties of network traffic, including Ethernet [51], MPEG [52] and JPEG [61] video, and WWW [82] traffic. Without doubt, fractional Brownian motion (FBM) for

exactly self-similar data traffic and fractional ARIMA for asymptotically self-similar video traffic, are better models to capture the slowly decaying correlation function and scale invariant burstiness.

The MWM derived in [44] gave even better results for modelling these types of traffic. However, there is a lot of debate about whether LRD matters in traffic management for delay-sensitive services. Being supported by [70], [75], [76], [77] and [78] it is argued that in a practical mission-critical control environment, only limited traffic time-scales is a concern. This implies that when the arrivals of a certain time instant are being served, arrivals with a time lag greater than the maximum permitted delay has no effect; packets having to wait longer than the maximum permitted delay are dropped. Therefore, ARMA or MMPP models can be used for real time traffic since their correlation in the form of summarized exponential functions, approximate the hyperbolically decaying correlation.

It is evident from previous work that researchers try to motivate why one specific network traffic model should be used for network simulation. This study considers two separate models which best capture the characteristics of TCP and UDP traffic respectively.

5.3 REAL TRAFFIC TRACES

Generating traffic for simulation first requires analyses of real traffic. It is important though to choose the right time, duration and source of data. The dimensioning to be performed applies to data entering a backbone network. For this reason various backbone traffic measurements were analyzed. Typical results are shown to reveal some important observations.

Measurements shown in figures 5.1 and 5.2 followed after the development of an IP monitoring (IPMON) system, described in [56]. IPMON was deployed on the Sprint IP backbone to improve analysis capabilities. IPMON, along with the Simple Network Management Protocol (SNMP), was used to capture traffic loads over various time intervals.

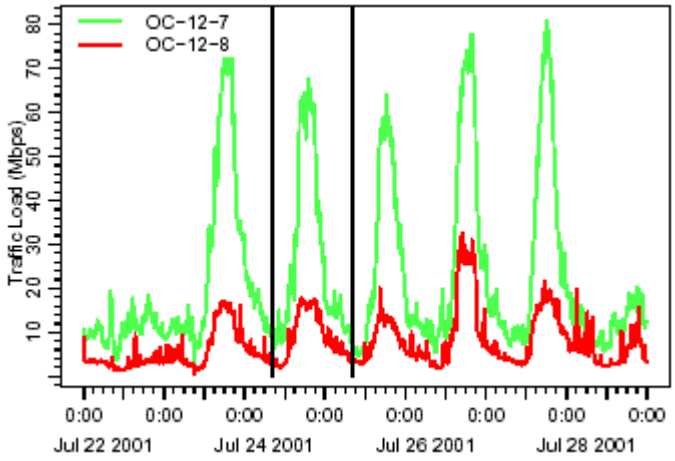


Figure 5.1 Week-long traffic trace measured on the Sprint IP backbone in July 2001 adopted from [56]

Two observations can be made from their results:

- Traffic load reported by SNMP (OC-12-8 trace) was lower than that from the IPMON (OC-12-7) measurements. This is because the SNMP statistic is an average over 5 minutes, while the IPMON measured traffic load is calculated in 1 second intervals. This shows that the traffic is more bursty in a finer time granularity. This is also confirmed in the work of [82]. In other words, SNMP statistics are not appropriate to detect short term congestion. In this research even smaller time intervals (as low as 10ms for RT analysis) have to be used since these directly affect the accuracy of the results. It is thus very important to use the most appropriate time granularity for analysis as well as synthesis.
- The second implication is that of the time duration of the trace to be used for analysis. Figure 5.1 shows traffic load for a week period, which exhibits large periodic bursts. The bursts are a direct result of the five working days of the week, with the low periods over the weekend. These characteristics are completely irrelevant with respect to the short term delay sensitive needs of most network applications. Figure 5.2 shows the same workload for a 24 hour period. In this case it is clearly evident that the highest workload occurs between 11am and 8pm with isolated bursts in the low period. QoS needs to be delivered at all times, especially in the peak workload periods. Consequently, only peak time traces are used in analysis for dimensioning tasks, as indicated in figure 5.2.

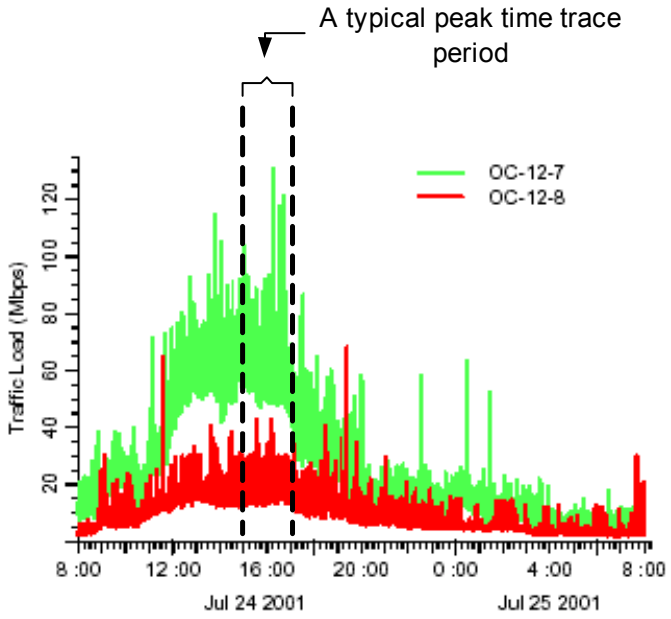


Figure 5.2 Day-long traffic trace measured on the Sprint IP backbone in July 2001 adopted from [56]

5.4 THEORETICAL CONSIDERATIONS

In chapter 2 it was clearly stated that modern integrated networks host a great diversity of applications each having their own QoS specifications. These applications are aggregated into traffic classes for distinctive handling, as described in chapter 3. Since the Real Time traffic class essentially hosts UDP traffic, while the rest of the classes host mostly TCP, the simulation will be performed accordingly. To perform accurate traffic modelling, various models were studied and explained in chapter 4. The evaluation of the traffic models is based on the queuing behaviour, correlation matching, and marginal distribution.

5.4.1 Queuing Behaviour

Queues (buffers) are widely used in modern telecommunication networks including switches, routers, servers, etc. They play an integral role in the performance of a network. They determine the amount of traffic a network can handle, the delays packets will suffer and the packet loss properties of the network. All of these are vital in modern telecommunication

networks and influence a network's ability to guarantee QoS. This is why the queuing behaviour is certainly the most important of all the experiments.

To evaluate the queuing behaviour of the traffic traces, the well-known, widely used infinite buffer experiment was used to determine the buffer overflow probability. Overflow properties are very important in networks because packets are lost when buffers overflow, resulting in serious network performance degradation.

This experiment makes use of a buffer of infinite length and a constant server rate. The data trace is then fed into the buffer while the buffer tail (Q), the amount of bytes in the buffer at each time instant, is monitored. Once the whole trace has passed through the buffer the overflow probability can be computed. That is the probability that a buffer of length x will overflow. This sample probability can be calculated for different values of x with the following formula:

$$P(Q > x) = \frac{N_Q}{N_T}, \tag{5.1}$$

where N_Q is the number of instances where $Q>x$, and N_T is the total number of time instances. For visual interpretation, these probabilities are plotted as $\log(P(Q > x))$ against x in Chapter 6. Another important aspect of the buffer overflow probability is that it will give an indication of the size of the buffers to be used in a network. With a certain probability it will ensure that buffers will not overflow. This is very useful when designing new networks.

5.4.2 Correlation Matching

The autocorrelation of traces is an indication of how well the model is capable of capturing long range dependency. This phenomenon is not necessarily present in all arrival processes, but can also be compared to original traces, to evaluate models.

5.4.3 Marginal Distribution

The marginal distribution shows the number of time units for which the bytes in the trace are within a certain size range. For example, the number of time units where the byte sizes, are between 100 and 200. Results of these tests are presented in Chapter 6.

5.5 TRAFFIC GENERATION

5.5.1 The Multi-fractal Wavelet Model

The Haar wavelet described in chapter 4 is used for the MWM. The authors of [44] proposed 3 constraints for the Haar wavelet with regards to the synthesis of Ethernet (mainly TCP) traffic. The three constraints and the fitting and synthesis procedure of [44] were followed as described below.

The Non-Negative Constraint

The Haar wavelet analysis is performed by using equations (4.17) and (4.18), and the synthesis is described by equations (4.19) and (4.20). Since each scaling coefficient, $c_{j,k}$, only equals a scaled local mean it is known that $c_{j,k} \geq 0$ is a sufficient condition for synthesizing non-negative data. Thus $c_{j+1,2k} \geq 0$ and $c_{j+1,2k+1} \geq 0$ lead to the following simple constraint [45] to guarantee a non-negative synthesized data trace

$$\left|d_{j,k}\right| \leq c_{j,k} \tag{5.1}$$

Note that this constraint is only applicable to the Haar wavelet transform. To implement the above constraint, a simple multiplicative wavelet model is introduced in [44]. The wavelet coefficients are computed recursively by

$$d_{j,k} = A_{j,k} c_{j,k} \tag{5.2}$$

with $A_{j,k}$ a random variable supported in the interval [-1,1]. Using (5.2) together with (4.19) and (4.20) results in

$$c_{j+1,2k} = \left(\frac{1+A_{j,k}}{\sqrt{2}}\right)c_{j,k} \tag{5.3}$$

and

$$c_{j+1,2k+1} = \left(\frac{1-A_{j,k}}{\sqrt{2}}\right)c_{j,k} \tag{5.4}$$

Observe that the wavelet coefficients have been eliminated altogether, but they are still inherently present through the multipliers, simplifying the model even more. The multipliers, $A_{j,k}$, are independent both between scales and within scales, resulting in the wavelet coefficients being dependent, but uncorrelated. This follows from the fact that if the assumption is made that $A_{j,k}$ is symmetric about 0, then $E[A_{j,k}] = 0$ [44].

Controlling the Correlation Structure

The authors of [44] observed that the multipliers only influence the correlation structure of the model and not the higher-order moments. The correlation structure of the synthesized data can thus be controlled by choosing the variances of $A_{j,k}$ to approximately match the variances of the wavelet coefficients between scales. The easiest way to control the energy scaling is to fix the energy at scale $j = 0$, and set the ratios between scales as

$$\eta_j = \frac{Var(W_{j-1,k})}{Var(W_{j,k})} \tag{5.5}$$

The variances used in (5.5) are the sample variances of the wavelet coefficients obtained during the wavelet analysis. In order to show that the variances can be expressed in terms of the multipliers $A_{j,k}$, the scaling coefficients must be expressed in terms of the coarsest scaling coefficient $U_{0,0}$, using the shift indexing described in Appendix B.

$$U_{j,k_j} = U_{0,0}\prod_{i=0}^{j-1}\left(\frac{1+(-1)^{k_i'}A_{i,k_i}}{\sqrt{2}}\right) \tag{5.6}$$

The wavelet coefficients can also be expressed in terms of $U_{0,0}$ and $A_{j,k}$ using the same approach, namely

$$U_{j,k_j} = A_{j,k_j}U_{0,0}\prod_{i=0}^{j-1}\left(\frac{1+(-1)^{k_i'}A_{i,k_i}}{\sqrt{2}}\right) \tag{5.7}$$

This is needed to express the ratio in terms of the energy of the multipliers using equations (5.5), (5.7) and the fact that $E[A_{j,k}]=0$. A long simplification process presented in [44] results in

$$\eta_j = \frac{2E\left[A_{j-1,k}^2\right]}{E\left[A_{j,k}^2\right]\left(1+E\left[A_{j-1,k}^2\right]\right)} \tag{5.8}$$

By calculating (5.5) during the analysis process and saving the ratios at each scale, the energy properties between scales of the multipliers can be determined. Using (5.8) together with the calculated ratios, the energy of $A_{j,k}$ at each scale can iteratively be calculated. The problem with this approach, as stipulated in [44], is to determine at which value the energy should be fixed at scale $j = 0$. The energy of $A_{0,k}$ can be determined by

$$E\left[A_{0,k}^2\right] = \frac{E\left[W_{0,0}^2\right]}{E\left[U_{0,0}^2\right]}. \tag{5.9}$$

But the problem is how to determine the expected values of the wavelet and scaling coefficients determined if there is not a large enough set of data points left at scale $j = 0$.

By assuming a particular distribution (to be discussed) for the wavelet and scaling coefficients at scale, the above-mentioned problem can be solved. The distribution should be strictly positive to ensure a non-negative data trace. The authors of [44] have chosen a Gaussian distribution, which is only characterized by its mean and variance. For this model to satisfy the non-negative condition, the mean must be much greater than the variance. For any distribution it is known that

$$E\left[X^2\right] = \mu^2 + \sigma^2, \tag{5.10}$$

where μ is the mean and σ^2 is the variance of the distribution. Combining (5.9) and (5.10) results in

$$E\left[A_{0,k}^2\right] = \frac{E\left[W_{0,0}\right]^2 + Var\left(W_{0,0}\right)}{E\left[U_{0,0}^2\right]^2 + Var\left(U_{0,0}\right)}, \tag{5.11}$$

which can be used to initialize the iterative process for synthesis.

Distribution of the Multipliers

The previous section shows that it is necessary to be able to control the variance of the multipliers at different scales. For this reason the symmetric β-distribution was proposed by [44]. The symmetric β-distribution has the probability density function (PDF)

$$P_A(x) = \frac{(x-a)^{p-1}(b-x)^{p-1}}{\beta(p,p)(b-a)^{2p-1}}, \qquad (a \le x \le b), \tag{5.12}$$

which is supported on $[a,b]$ and where $\beta(p,p)$ is the beta function and $p > 0$ is the shape factor. For this particular use the distribution has to be supported on $[-1,1]$ which yields the following PDF

$$P_A(x) = \frac{(x+1)^{p-1}(1-x)^{p-1}}{\beta(p,p)2^{2p-1}}, \qquad (-1 \le x \le 1) \tag{5.13}$$

and variance function

$$E\left[A^2\right] = Var(A) = \sigma^2 = \frac{1}{2p+1}. \tag{5.14}$$

The variance of the symmetric β-distribution is matched with the variance ratios calculated with (5.8). Using (5.8) and (5.14), the values of the p 's to match the desired variance structure can be calculated using

$$p_j = \frac{\eta_j\left(p_{j-1}+1\right)-1}{2}. \tag{5.15}$$

Using (5.11) together with (5.14) the iterative process can be initialized with

$$p_0 = \frac{E\left[U_{0,0}^2\right]}{2E\left[W_{0,0}^2\right]} - \frac{1}{2} \quad . \tag{5.16}$$

Fitting and Synthesis Procedure

The following steps, as described in [44], are followed to fit and synthesize the MWM to actual data traces.

1) A wavelet analysis of the actual data trace is first performed using the Haar wavelet transform as explained in equations (4.17) and (4.18). The energy ratios are calculated using (5.5) at each scale during the analysis. The number of scales, n, used in the analysis is very important since it is needed to estimate $E\left[U_{0,0}^2\right]$ and $E\left[W_{0,0}^2\right]$ from the coarsest scale coefficients. For these estimates to be reliable, enough wavelet- and scaling coefficients are needed after scale *n analysis.*

2) Set $j = 0$ at the coarsest scale and calculate the shape factor, p_0, of the β- distribution with (5.16).

3) Generate the β-distribution random multipliers $A_{j,k}$, with shape factor p_j at scale j.

4) Use $A_{j,k}, U_{j,k}$, (5.3) and (5.4) to calculate the scaling coefficients $U_{j+1,2k}$ and $U_{j+1,2k+1}$ at scale j.

5) Calculate the shape factor for the next scale, p_{j+1}, using the previously calculated ratios, η_{j+1}, and equation (5.15).

6) Iterate steps 3, 4 and 5, replacing j by $j + 1$ until $j = n$ is reached.

Once the finest scale, $j = n$, is reached the synthesis process is stopped. The scaling coefficients at scale $j = n$, are then used as the output of the model.

5.5.2 The Markov Modulated Poisson Process

In order to generate UDP traffic for the RT class, various UDP packet traces were analyzed. The mean arrival rate, the burstiness, and the ON-OFF probability were measured. To generate a packet trace the following steps are performed:

1) The ON-OFF probabilities are used to generate a series of ON and OFF intervals (two-state Markovian process).
2) For each ON interval, 1 of 3 Poisson processes (3-state Poisson) is triggered according to assigned probabilities.
3) The number of packets for the ON period are generated by means of a Poisson distribution (hence Markov Modulated Poisson Process). Each of the 3 Poisson states has a unique average that is derived from the real data traces.
4) In order to provide the node under simulation with the actual number of bytes arriving in each interval, each packet arrival, determined by the MMPP, is multiplied by a random process corresponding to the PSD of UDP traffic.
5) The mean, burstiness, and ON-OFF probabilities of the generated trace are measured again to compare with the original trace.

To substantiate the use of this model to represent UDP traffic, the queuing behaviour, the marginal distribution and the autocorrelation function for generated traces are compared to the same measurements of the real traces. Results are presented in Chapter 6.

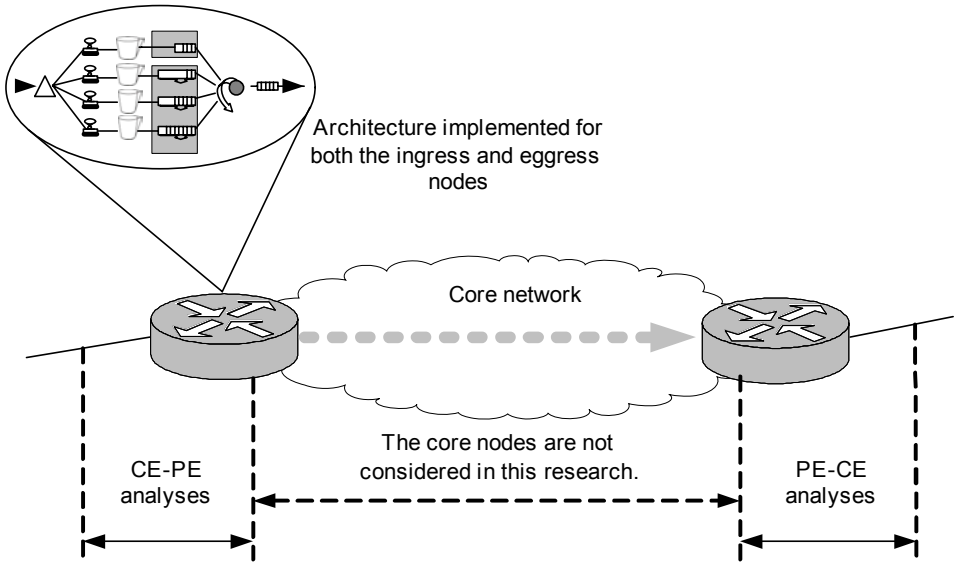
5.6 THE SIMULATION TEST BED

The first simulations were performed at the Telkom SA Ltd laboratory. The laboratory consists of a good real network environment providing numerous CISCO routers with the applicable software and hardware configurations. The AdTech [83] simulation tool was provided to generate class-based traffic. The AdTech tool is known for its very precise and accurate measurements. The only drawback of this setup was the limitation of the available traffic models provided by AdTech.

A large portion of this research covers, and states, the importance of accurate traffic modelling. The use of the latest traffic models described in academic literature was not feasible with AdTech. Hence other simulation environments capable of hosting the appropriate traffic models were required. Therefore, Matlab v6 Release 12.1 was used for the following reasons:

- versatile simulation environment,
- diverse statistical and mathematical toolboxes,
- flexible use of matrixes and arrays, and
- convenient logging, storing, and plotting of results.

The complete test bed as implemented in Matlab v6 R12.1 is depicted in figure 5.3. Note that only the edge provisioning is considered and hence only the edge nodes are analyzed. Delay and loss budgets drawn up by CISCO [84] and the ITU specify how these are allocated to various parts of the network. Not all the parts contribute to the total delay budget, as illustrated in figure 2.2 (chapter 2), are included in the analysis. Later in this chapter these portions are tabulated to specify exactly how big the budget is for the edge nodes.



CE – Customer Edge Node

PE – Provider Edge Node

Figure 5.3 Simulation test bed

Vendors cater for customer requirements by structuring different combinations of traffic classes in packages. Table 5.1 lists the bandwidth distribution of a few packages available from Telkom on their VPN Supreme product.

Table 5.1 Example packages of Telkom's VPN supreme product.

Package	Access pipe	RT	IB	BB	GD
Debut 2	128	0	0	0	128
Foundation 3	256	0	0	256	0
Meridian 12	1984	0	512	0	1472
Converged 2	128	32	0	0	96
Converged 22	512	128	64	320	0

From table 5.1 it is evident that separate class analysis is necessary to cater for packages consisting of only some of the traffic classes. Hence simulation is performed separately for each class.

5.6.1 Single Class Simulation

Figure 5.4 illustrates a node for the single class analysis.

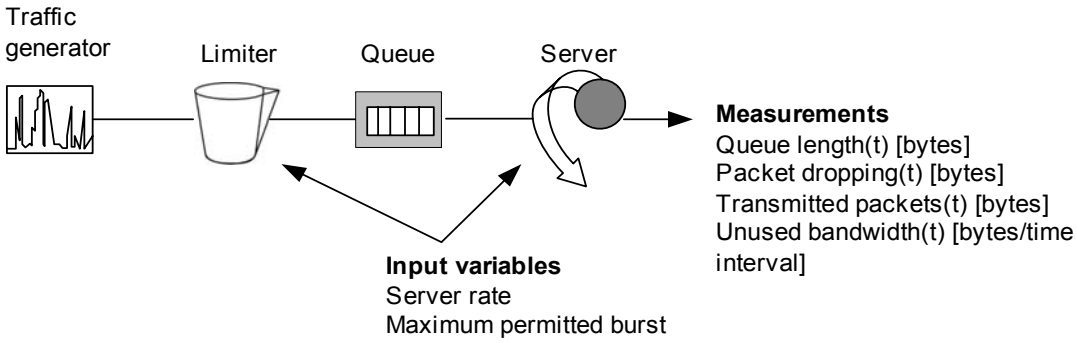


Figure 5.4 Single class simulation setup

In the case of the RT traffic class the following components are used:

- The MMPP is used as traffic generator.
- The policer is used as rate limiter: Out-of-bound packets are discarded.
- The queue is FIFO.
- RT traffic is concerned as the priority class in LLQ; consequently the arriving traffic has full access to all the allocated bandwidth.

In case of the other three classes the following components are used:

- The MWM is used as traffic generator
- The shaper is used as rate limiter: Out-of-bound packets are delayed.
- All the queues itself are FIFO queues.
- These traffic classes share the CBWFQ discipline in LLQ. However, in the single class analysis, the incoming traffic has access to the full allocated bandwidth for the concerned class.

The single class simulation includes the following steps:

1) The input parameters: provisioning factor and maximum permitted burst are assigned start, stop and resolution values.
2) A complete trace is generated and stored in a matrix by:
    a. Generating the number of arrivals per sampling period according to the applicable traffic model.
    b. Using the appropriate PSD to generate a packet size for each arrival.
    c. Accumulating the total number of bytes for each sampling period.
3) The average data rate of the trace is multiplied with the provisioning factors in order to determine the server rate.
4) The complete traffic trace is sent through the node architecture for each combination of the specified input parameters (sweeping two-dimensionally).
5) The four specified measurements are logged as a function of time.
6) The unused bandwidth is used in the multi-class simulation to share with lower priority classes.
7) At the end of each trace the percentage loss and the cumulative delay distribution are determined and plotted as a function of the input variables.
8) From these graphs, the minimum values of input variables resulting in adequate QoS measurements are logged.

Simulations are performed 50 times for each class. The results are presented in the form of histograms and graphs in Chapter 6. Average and boundary values for provisioning factors and maximum permitted bursts are provided.

5.6.2 Multi-Class Simulation

The single class analysis does not include the effects of bandwidth sharing and statistical multiplexing. In order to analyze these effects, the combination of all the classes has to be simulated. Figure 5.5 illustrates the multi-class simulation setup.

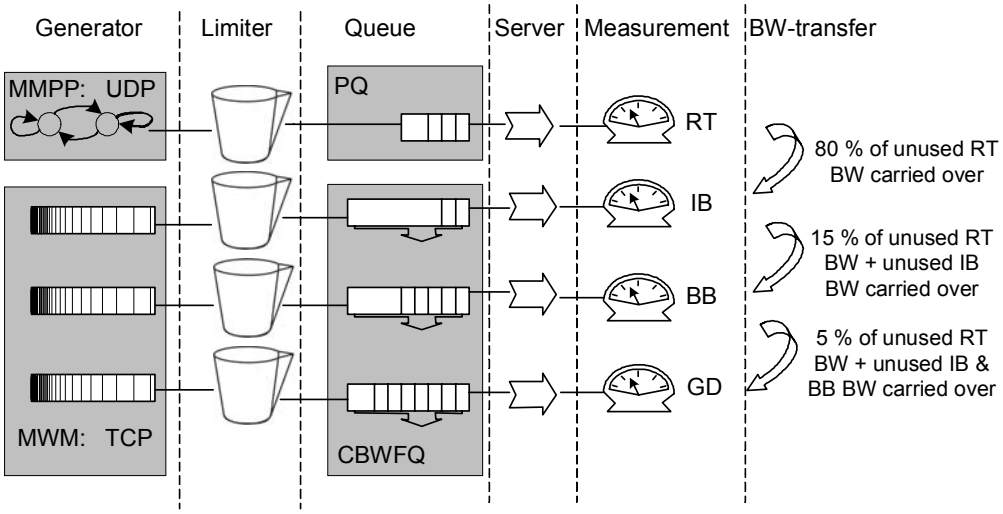


Figure 5.5 Multi-class simulation setup

The multi-class simulation includes the same steps as the single class, except for the following:

1) The bandwidth allocated to the combination of the 4 classes is determined by the results of the single class simulation and the dimensioning method described in chapter 3.
2) The bandwidth available for each class is determined by the nature of the LLQ scheduling discipline. I.e. if any packets are waiting in the RT queue, they will be transmitted first (the RT class is bounded by the policer). If the RT queue is empty the available bandwidth is shared on a weighted basis as shown in figure 5.5 (80%, 15%, and 5%).

The multi-class simulation determines if the dimensioning values obtained by the single class simulation, perform equally well in an environment exposed to bandwidth sharing and statistical multiplexing.

QoS parameter budgets for specific network portions are defined by Cisco [84] and the ITU. In Chapter 6, the requirements described in the budgets are used to read the sufficient dimensioning values from plotted output data. The next section covers these requirements.

5.7 QOS REQUIREMENTS

The following classes of services and associated QoS metrics with end-to-end worst case performance bounds are defined by Telkom [85]. These values are based on recommendations by the ITU-T, Cisco networking and Juniper networks.

Table 5.2 Per Class performance bounds

Class of Service	Delay	Loss	Jitter
Real-time	65ms OW	1%	Minimized
Interactive Business Data	250ms RTT	0.25%	N/a
Bulk Business Data	400ms RTT	2%	N/a
General Data	N/a	N/a	N/a

RTT = Round Trip Time, OW = One Way

Other vendors define their own service classes, but the same principles apply. The question is not which classes, or how many, are defined by a vendor, but rather which applications are to be included in the concerned class. It is the type of applications belonging to a class that determines its performance bounds.

5.7.1 Delay Budgets

When optimizing resource dimensioning, it is important to consider the end-to-end delay budgets of the different classes. Even though a SLA stipulates certain end-to-end delay requirements, the various portions of the network have different requirements. The following section highlights the different delay budgets considered for each class.

Real Time

For a managed voice service, a mouth-to-ear delay of 150ms is considered good (based on ITU Recommendation G.114), but delays of up to 200ms have been suggested as being acceptable [84].

Table 5.3 Mouth to ear delay

Element	Delay (ms)
Terminal-CE (Local + Distant)	30
Voice Codec (approx.)	40
Real Time CE-CE	65
De-jitter buffer	55
Mouth-to-Ear (Total)	190ms

Comments:

- On both ends of a connection 15ms is budgeted for delay from the terminal (e.g. IP Phone) to the CE at a customer site.
- The 40 ms delay specified for the voice codec includes serialization delay in the customer portion of the network, and is based on coders including ADPCM, G.723, G.726, G.729 [9].
- To fit into a delay budget of 190ms, delays of 65ms and 55ms are allocated to the provider core portion of the network and the de-jittering buffers respectively, on both ends.
- The maximum jitter expected from a 65ms delay is 55ms CE-CE[8]:
    - Maximum jitter is determined by the largest variable delay component in the CE-CE portion (see table 5.4).
    - Table 5.4 excludes the worst case PE-PE[9] propagation delay but includes the serialization delays in the access links.
- ITU Recommendation G.131 requires that echo cancellers be used when the mouth-to-ear delay exceeds 25ms.

[8] CE-CE – Customer Edge node to Customer Edge node. It includes these two nodes and the whole provider portion of the network. Refer to section 3.5 for a detailed description.

[9] PE-PE – Provider Edge to Provider Edge. This includes only the provider portion of the network.

Table 5.4 Real Time Delay Budget

Element	Delay (ms)
PE-PE Propagation (1500km)	10
PE-PE Queuing	15
CE-PE (Local + Distant)	40
CE-CE (Total)	65ms

Comments:

- Propagation delay cannot be changed. Within South Africa a maximum of 10ms (1500km) is generally assumed.
- 25ms is dedicated for the PE-PE delay (Propagation and queuing combined).
- Core queuing is small relative to edge queuing due to fast switching: 5ms per hop in the provider portion (assuming max. 3 hops), excluding propagation delay.
- Provider edge node traffic conditioning: 20ms per access circuit, including serialization delay.

Interactive Business Data

The SLA for the Interactive Business Data class (defined by Telkom) is 250ms Round Trip. This translates to a 125ms one-way delay requirement.

Table 5.5 Interactive Business Data Delay Budget

Element	Delay (ms)
PE-PE Propagation (1500km)	10
PE-PE Queuing	30
CE-PE (Local + Distant)	80
CE-CE (Total)	120ms

Comments:

- 40ms is dedicated for the PE-PE delay (Propagation and queuing combined).
- 10ms is dedicated per hop in the provider portion (assuming max. 3 hops), excluding propagation delay.
- 40ms is dedicated per access circuit, including serialization delay.

Bulk Business Data

The SLA for the Interactive Business Data class is 400ms Round Trip. This implies a 200ms one-way requirement.

Table 5.6 Bulk Business Data Delay Budget

Element	Delay (ms)
PE-PE Propagation (1500km)	10
PE-PE Queuing	60
CE-PE (Local + Distant)	130
CE-CE (Total)	200

Comments:

- 70ms is dedicated for PE-PE (Propagation and queuing combined).
- 65ms is dedicated per access circuit, including serialization delay.
- 20ms is dedicated per hop in the provider portion (assuming max. 3 hops), excluding propagation delay.

General Data

Since no mission critical data belongs to this class no tight bounds are defined.

5.7.2 Discussion

As stated in chapter 2, the main contributor to the total delay that can be controlled, is the queuing delay. The other delays are inherent to physic processes, and the way routing and switching is done. However, it is important to keep in mind that serialization delay becomes very significant when very low data rates are analyzed. Table 5.7 lists examples of serialization delays for various line speeds and frame sizes.

Table 5.7 The serialization delays corresponding to typical frame sizes and line speeds.

Frame Size (bytes)	Line Speed (Kbps)										
	19.2	56	64	128	256	384	512	768	1024	1544	2048
38	15.83	5.43	4.75	2.38	1.19	0.79	0.59	0.40	0.30	0.20	0.15
48	20.00	6.86	6.00	3.00	1.50	1.00	0.75	0.50	0.38	0.25	0.19
64	26.67	9.14	8.00	4.00	2.00	1.33	1.00	0.67	0.50	0.33	0.25
128	53.33	18.29	16.00	8.00	4.00	2.67	2.00	1.33	1.00	0.66	0.50
256	106.67	36.57	32.00	16.00	8.00	5.33	4.00	2.67	2.00	1.33	1.00
512	213.33	73.14	64.00	32.00	16.00	10.67	8.00	5.33	4.00	2.65	2.00
1024	426.67	149.29	128.00	64.00	32.00	21.33	16.00	10.67	8.00	5.31	4.00
1500	625.00	214.29	187.50	93.75	46.88	31.25	23.44	15.63	11.72	7.77	5.86
2048	853.33	292.57	256.00	128.00	64.00	42.67	32.00	21.33	16.00	10.61	8.00

From table 5.7 it can be seen that serialization delay becomes large at low line speeds. A typical RT packet (like voice) consists of 64 bytes, which will results in a delay of 26.67ms if the line speed is 19.2 kbps. If this time is accumulated for the ingress[10] and egress[11] nodes, it results in 53,3ms, which already exceeds the combined delay budget for serialization delay

[10] Ingress node – The first node as the packet enters a particular domain.

[11] Egress node – The last node when a packet exits a domain.

and queuing delay. This situation is very undesirable. For mission critical data it is thus important to realize that a very low demand cannot be catered for solely by means of over provisioning. This statement also applies for simulation purposes, and hence relative large traffic loads will be considered.

5.7.3 Loss Budgets

Loss budgets are merely percentages of lost bytes or packets relative to the total amount. The 1% for Real Time, 0.25% for Interactive Business and 2% for Bulk Business as depicted in table 5.2 will be used.

5.7.4 Jitter Budgets

As observed in table 5.2, jitter is only relevant with regards to the Real Time class. In this case it is 55ms. This constraint will not be included in simulations since it is highly unlikely that the jitter will reach such high values. If it does, the applicable dimensioning values will already be affected due to the delay constraints.

5.8 OVERHEAD CONSIDERATIONS

Two forms of overhead need to be considered, i.e. Layer 1 and Layer 2 overhead as discussed in Chapter 3. For the purpose of overhead computation, this study assumes ATM and SDH as Layer 1 and Layer 2 protocols. With reference to Chapter 3, the combination of these layers results in 12.5% overhead. Secondly, management traffic is also transmitted on the same link. Management traffic is included in the Interactive Business class and considered to be 10% of the total bandwidth. The combination of lower layer and management overheads for the Interactive Business class results in 23.75% overheads. The rest of the classes will experience a 12.5% overhead. These overhead figures will be added to the end values.

5.9 CHAPTER SUMMARY

In this chapter the need to analyze real traffic traces is motivated and the particular traces used for this purpose are described. Theoretical considerations with regards to the concerned traces

are mentioned to support the choices for traffic models to be used in simulations. The steps that will be followed to generate traffic traces are described for the MWM and the MMPP respectively. The simulation test beds for single as well as multi-class analysis are described. Single-class analysis is described to find dimensioning values without the effect of statistical multiplexing. Multi-class analysis is described to evaluate the results obtained from the single class simulation under the effect of statistical multiplexing. The particular QoS parameters to be obtained for each class are stipulated. The chapter concludes by stating how much bandwidth should be dimensioned additionally to cater for lower layer overheads.

CHAPTER 6
RESULTS

6.1 REAL DATA

The-well known hour long packet trace (LBL-PKT-4) acquired at the Lawrence Berkeley Laboratory [86] was used for analysis. Separate recordings were made for UDP and TCP. The trace ran from 14:00 to 15:00 on Friday, January 21, 1994. The trace contains packet information consisting of a timestamp, (renumbered) source host, (renumbered) destination host, source port, destination port, and number of data bytes.

The traces were converted to contain only the time instances for each packet arrival. Sampling periods of 6ms were used to measure how many packets arrive during each time interval. Figure 6.1 shows the two arrival traces consisting of 2^{16} sampling periods for TCP and UDP respectively. These traces were used to fit the models, and are further referred to as the LBL-TCP and LBL-UDP trace.

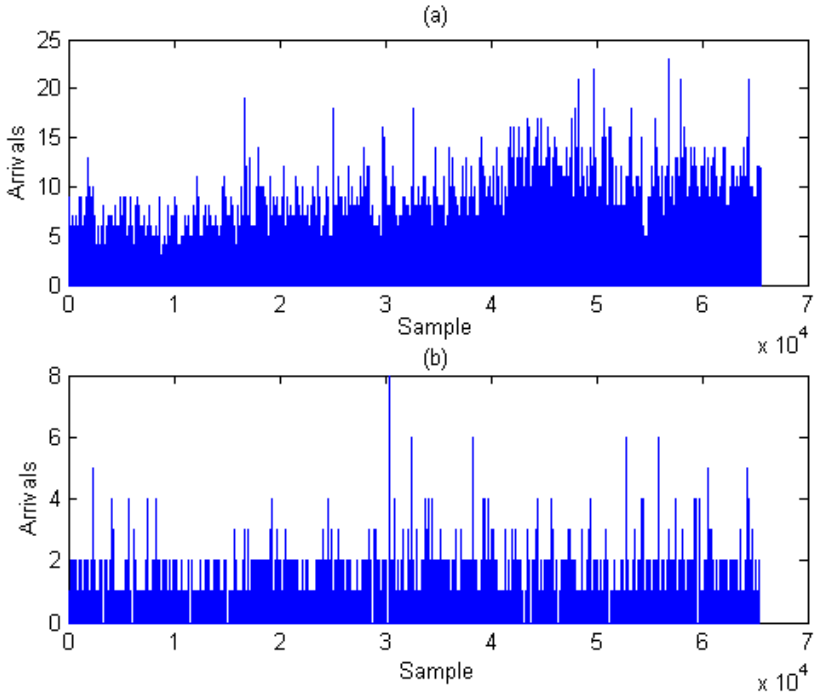


Figure 6.1 The LBL traces for (a) TCP and (b) UDP

Various sample sizes, ranging between 2^{13} and 2^{18} samples, were analyzed and synthesized to evaluate characteristic differences between trace lengths. There were no significant differences. Traces of length 2^{15} were generated for most simulations to limit the running-time.

6.2 TRAFFIC MODEL EVALUATION

Extensive work in [44] showed that the MWM is currently best for modelling Ethernet traffic. However, these Ethernet traffic traces can only be representative of TCP since nearly 95% of it consists of TCP packets [57]. An appropriate model was required for UDP. Closely following the work in [44], the MWM was tested to synthesize separate UDP traffic traces. It resulted in a slightly weaker approximation than the TCP synthesis in terms of the queuing behaviour, correlation matching, and marginal distribution. When comparing the synthesized trace with the original LBL-UDP traffic trace by mere inspection, it was found that two crucial characteristics of the synthesized trace did not correspond with the real trace:

- Occasional long off-periods present in the real UDP trace were not present in the synthesized trace.
- Although the average data rates over the complete traces corresponded, the burstiness of the synthesized trace was much higher (sometimes almost by a factor of 2) than that of the original trace.

The lack of correspondence with the off-periods motivated the use of an ON-OFF source. A few of Markov modulated processes were evaluated. The MMPP was eventually chosen based on the following results:

- The statistical descriptors presented in section 6.2 showed reasonable correspondences.
- The on-and off-periods were perfectly matched since the ON-probability and OFF-probability were derived from real traces.
- The mean arrival rate gave a perfect match over the entire trace, and additionally the mean is easily controllable.
- The burstiness showed a close correspondence (within 10%).

The burstiness of an arrival process is the key characteristic when queuing behaviour is considered. Delay, jitter and loss are primarily affected by the burstiness.

After analysis was done the MMPP was chosen to synthesize arrival processes in the case of UDP traffic, and the MWM was used for TCP. The appropriate PSDs were used in each case to complete the traces to byte level. To legitimize the use of these models, their queuing behaviour, autocorrelation function and marginal distribution are compared with those of the LBL traces.

6.2.1 Queuing Behaviour

Queuing behaviour is the most important property to match since it has a direct effect on most of the QoS parameters. The queue overflow experiment described in Chapter 5 was performed 20 times for each model. The averages for these experiments, compared with the queue overflow probability of the real traces are shown in figures 6.2 and 6.3.

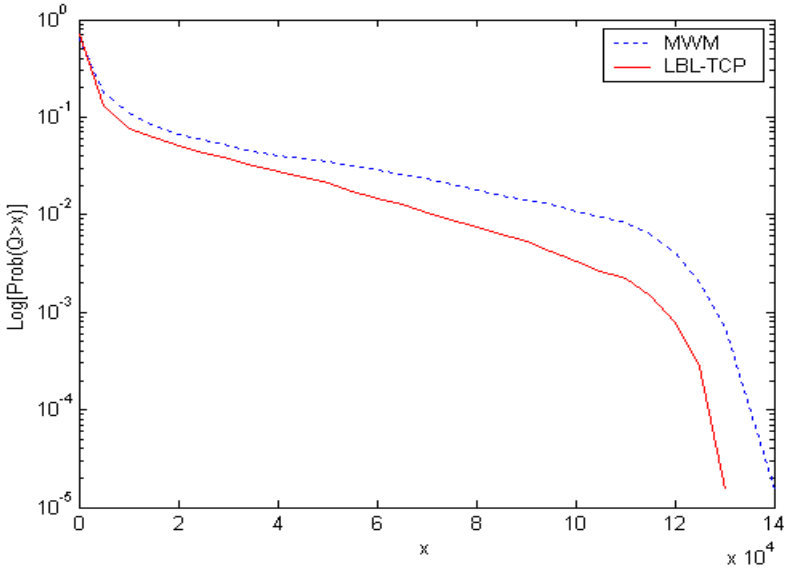


Figure 6.2 Queue overflow probabilities for the MWM and the TCP trace

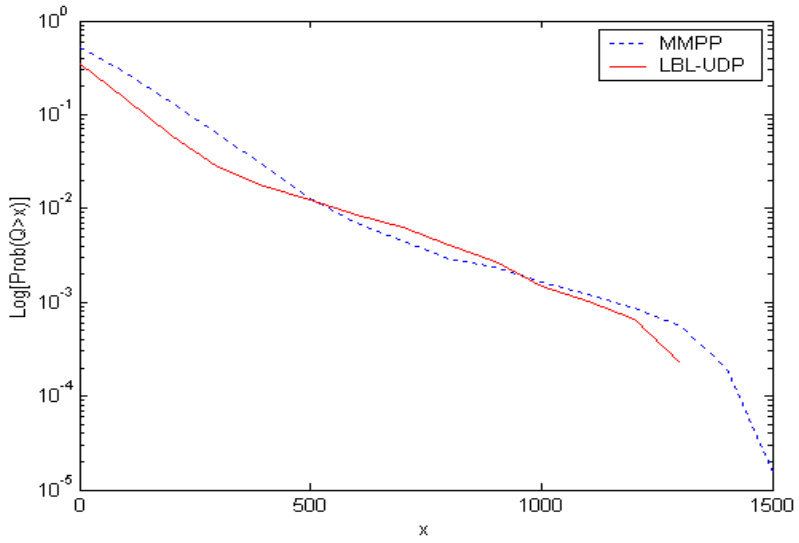


Figure 6.3 Queue overflow probabilities for the MMPP and the UDP trace

These results show that the queue overflow probabilities of the synthesized traces closely match those of the real traces. In case of the MWM, it shows an even more pessimistic result than the real trace, which provides a good margin for extreme bursts. These results confirm that the described models perform well in terms of their queuing behaviour.

6.2.2 Autocorrelation

Figure 6.4 presents the autocorrelation function for different time lags; the LBL-TCP trace in (a) and a MWM trace in (b).

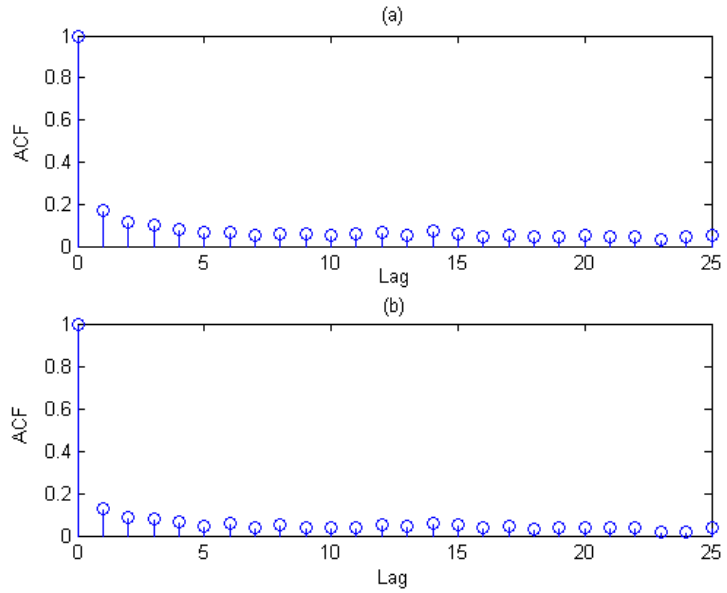


Figure 6.4 Autocorrelation function for (a) the LBL-TCP trace, and (b) a MWM trace

These traces show that little long range dependency actually occurs in the traces. More importantly, the two correlation functions closely resemble each other. This confirms that the long range dependency in the generated traffic corresponds to real traffic. The correlation structures for the UDP traces are depicted in figure 6.5.

In figure 6.5 it can be seen that the UDP trace shows little correlation for most of the time lags in consideration. The MMPP's inadequacy to capture long range dependency is clearly visible. But as illustrated in figure 6.5, there is no real need to capture such a characteristic, as it is not present in the real trace. This result was obtained from an analysis with a sampling period of 6ms. Using a sampling period of 100ms, long range dependency was indeed observed in the traces. However, for the RT traffic class only fine time scales are of concern. With regards to correlation matching, both models provide sufficient results.

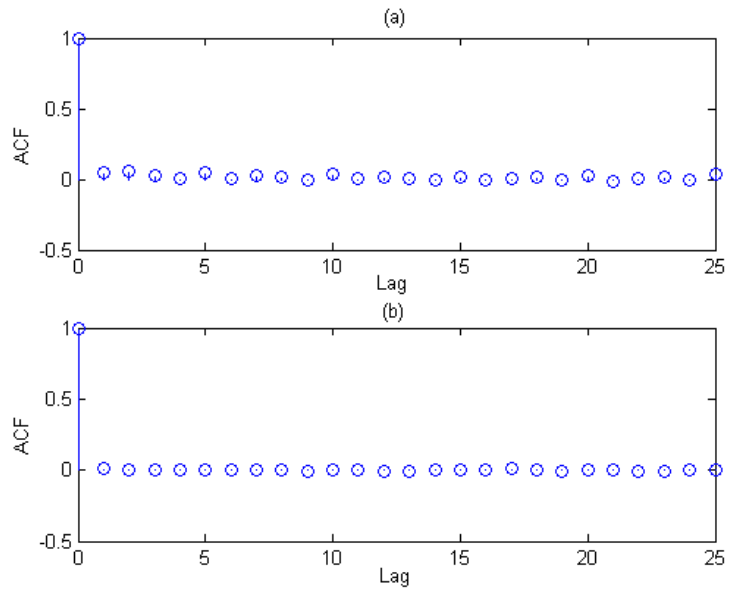


Figure 6.5 Autocorrelation function for (a) the LBL-TCP trace, and (b) a MMPP trace

6.2.3 Marginal Distribution

The comparison of the marginal distribution between the LBL traces and the MWM and MMPP traces are presented in figure 6.6 and 6.7 respectively. A logarithmic scale is used due to the very small number of time units hosting large numbers of bytes per time unit.

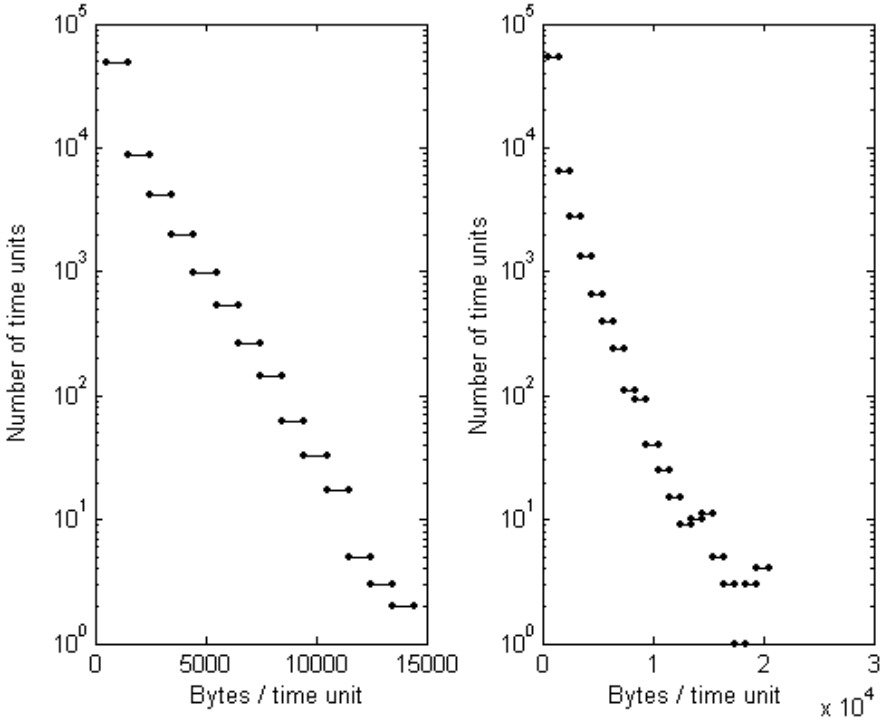


Figure 6.6 Marginal distributions for the LBL-TCP and MWM traces

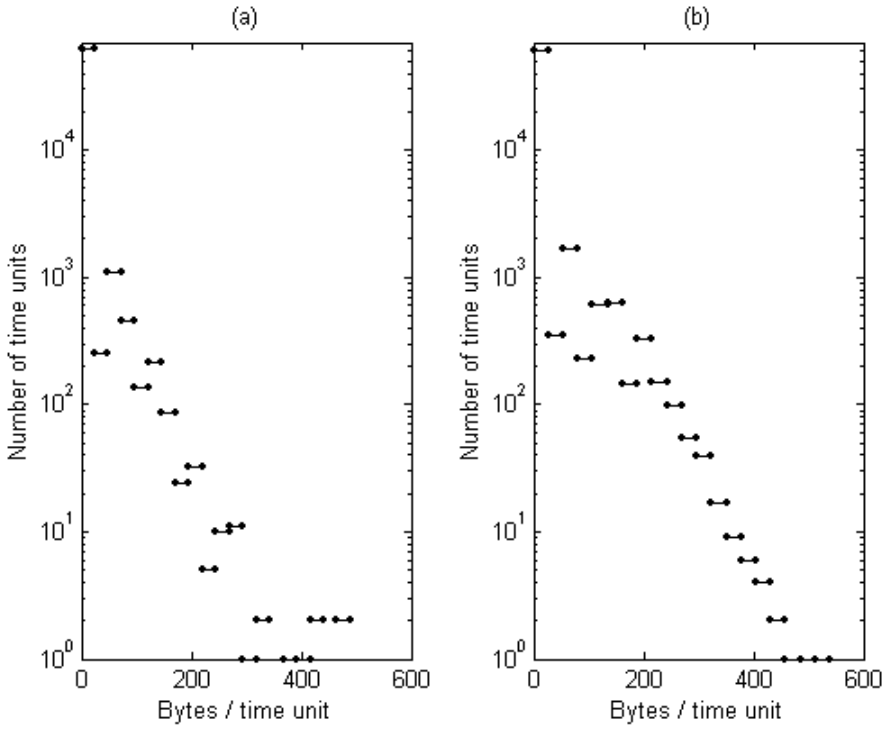


Figure 6.7 Marginal distributions for the LBL-UDP and MMPP traces

The marginal distributions for both UDP and TCP show a close correspondence. The results in this section confirm that the evaluated models are adequate for simulation purposes in their respective classes. For a more detailed description of the evaluation of traffic models, including the MWM, refer to [44]

6.3 SINGLE CLASS SIMULATION

The simulation test bed described in section 5.6 was used for single class simulations. During simulation, the *Provisioning Factor* (*PF*) and *Maximum Permitted* Burst (*MPB*), hereafter referred to as the *input variables*, are varied simultaneously. The purpose of these simulations is to find adequate input variables that result in the required delay and loss specifications for each class. This simulation does not include the effect of statistical multiplexing and dynamic bandwidth sharing. These effects are included in the multi-class simulations. It is performed only for the RT, IB and BB classes on a separate basis. GD is a best effort service and hence a provisioning factor of 1 is used. Delay and loss encountered at each sampling interval were logged against the corresponding input variables.

Delay is computed as a function of the server rate and the queue length. The queue length increases with an amount equivalent to the arrivals for each sampling period. The queue length also decreases with the number of bytes that will be transmitted during the same sampling period. For each sampling period the queue length is measured. With the server rate known, the delay is computed. At the end of a complete trace, the 99% cumulative delay is used for further analysis. The reason for using this delay measure is to eliminate poor average measurements, but at the same time exclude rare extreme-high delays. When the maximum delay is evaluated, occasionally a single very high delay results in over-dimensioning. This, again, unnecessarily leads to poor utilization. Figure 6.8 shows a typical cumulative delay, measured for the RT class with an indication of the 99%.

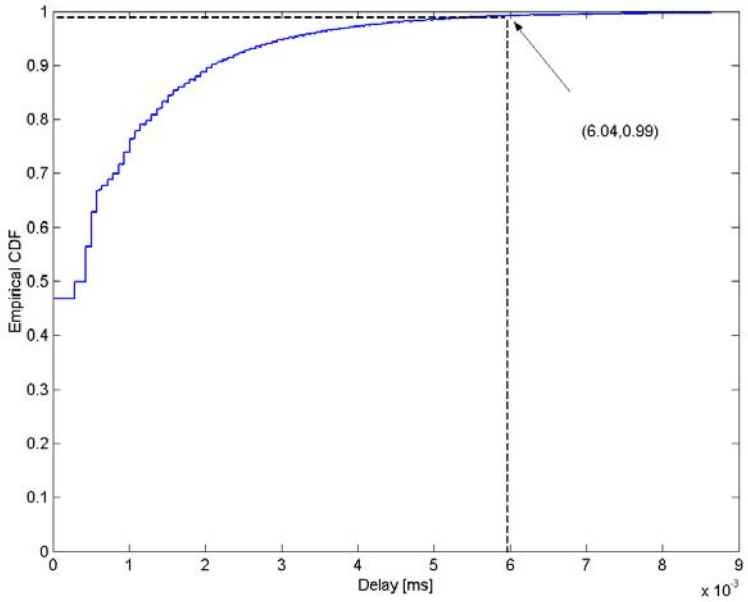


Figure 6.8 Typical cumulative delay distribution for the RT class

The 99% cumulative delay measure and the total loss are plotted as a function of the provisioning factor and the maximum permitted burst. These results are presented in the following section.

Losses are counted when queue lengths grow larger than the maximum permitted burst size. At the end of a complete trace the loss encountered at the ingress node is added to the loss encountered at the egress node. The total loss is presented as a percentage of the total number of bytes lost during the simulation of a complete trace.

6.3.1 The Effect of *PF*

Figure 6.9 illustrates how the delay and loss are affected by the Provisioning Factors (*PF*). Each of the line styles indicates a particular Maximum Permitted Burst (*MPB*).

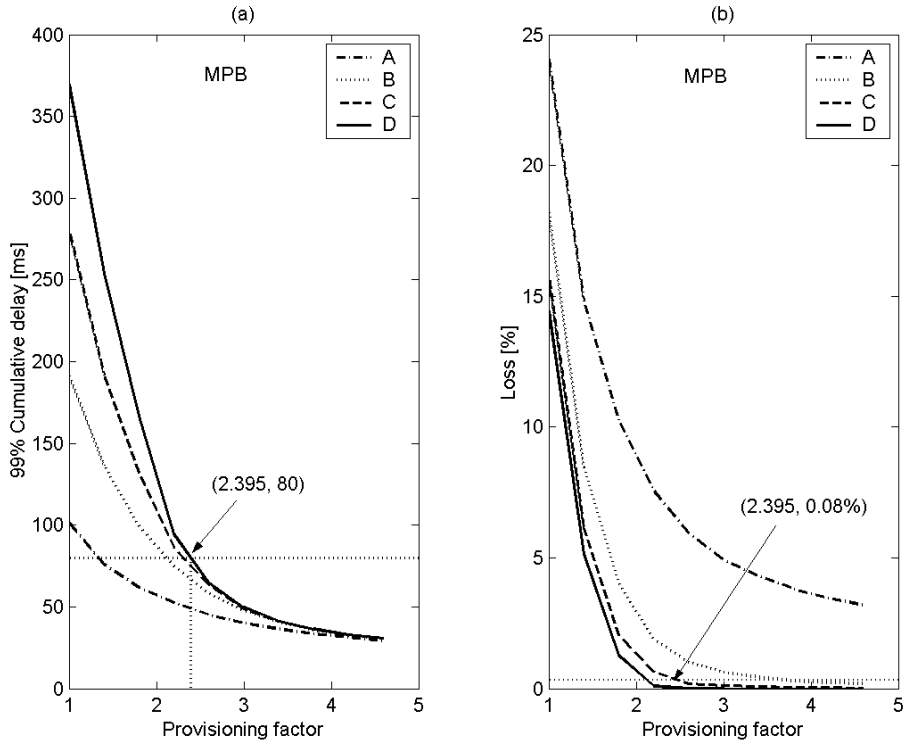


Figure 6.9 Delay and loss measurements as a function of PF

In figure 6.9, (a) depicts the delay vs. *PF* for 4 different *MPB* sizes. In (b), the loss is plotted against the *PF*. For illustration purposes, only 4 different *MPB* sizes were used. Throughout the simulations, a much higher resolution was used, typically between 10 and 20. Lines A to D represent the four different *MPB* sizes, where $A < B < C < D$.

To serve as an example, the IB class QoS parameters are used to illustrate how the graphs were used to accumulate *PF* readings. The required parameters for the IB class, as discussed in Chapter 5, are a loss bound of 0.25% and a delay bound of 80ms. From the graph, it is easy to observe the corresponding values for the *PF* and *MPB*. From (a) it can be seen that any of the *MPB* values could be used to result in a delay of less than 80ms. Line A would result in the lowest *PF* of approximately 1.3. However, this value can not be accepted as it would result in an expected loss of 18% as indicated by (b). Similarly, lines B and C result in loss values of 2.5% and 0.6% respectively. In this case only D satisfies both delay and loss requirements. As a result, the *PF* is chosen as 2.395 with a *MPB* of $4x$, where x can be any value related to the

data rate. From this result, it is clear that the measured delay as well as loss decreases exponentially with an increase in *PF*.

6.3.2 The Effect of *MPB*

Figure 6.10 illustrates how the delay and loss are affected by the *MPB*. Each of the line styles indicates a particular PF.

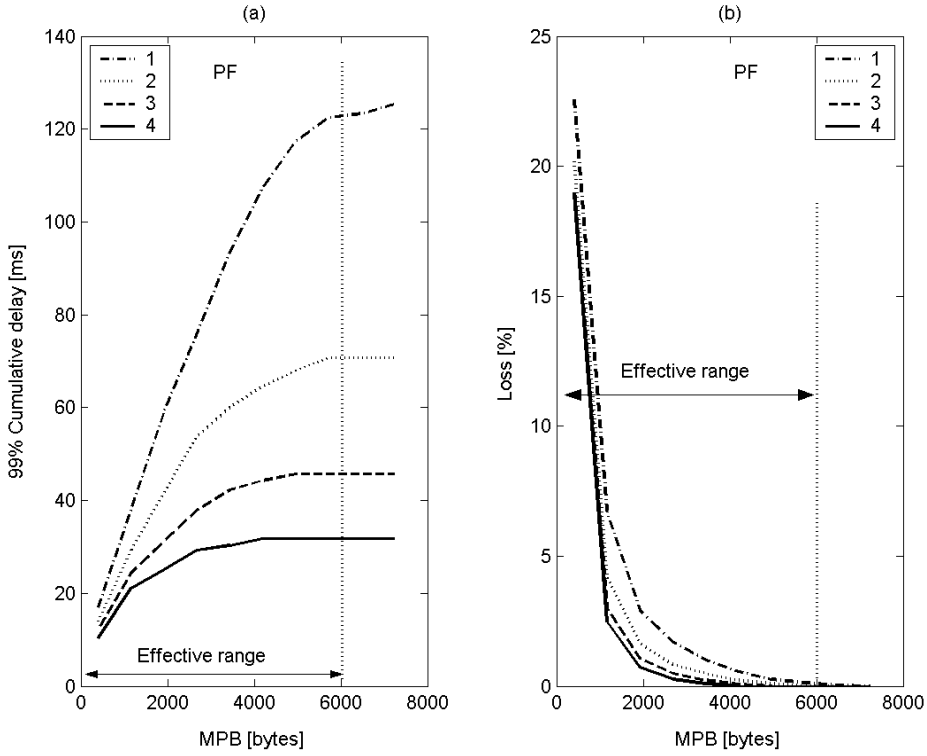


Figure 6.10 Delay and loss measurements as a function of MPB

Results show that there exists a certain range of *MPB* values that have a strong effect on the delay and the loss, hereafter referred to as the *effective range*. In the example of figure 6.10 the effective range exists from 0 to approximately 6000. Recall that *MPB* controls the queue length. For small *MPB*, the delay approaches zero while the loss approaches 100%. This can be compared to a network node without a buffer. On the other end of the effective range, where *MPB* exceeds 6000, the delay becomes stationary at a certain value, and the loss becomes zero. This situation can be compared to an infinite buffer without a token bucket

implemented. Numerous simulations have proven that the effective range is directly related to the average arrival rate and burstiness of the arrival process.

Within the effective range, an increase in *MPB* results in lower loss, but higher delay due to longer queue lengths. Loss reduces because of smaller overflow probabilities. For each class, the simulation results are used to determine which input variables work best to obtain the particular delay and loss measures specified for each class.

6.3.3 The Effect of the Input Variables on Utilization

As stated in Chapter 1, the objective is to optimize bandwidth utilization while maintaining the required QoS. Figure 6.11 shows that the utilization of allocated bandwidth decreases exponentially with a growing *PF*. The utilization improves with an increase in *MPB*, but only within the effective range. This is due to loss. When the *MPB* exceeds the effective range, loss measures cannot decrease beyond zero. From the point where the loss reaches the top end of the effective range, an increase in *MPB* does not have any effect.

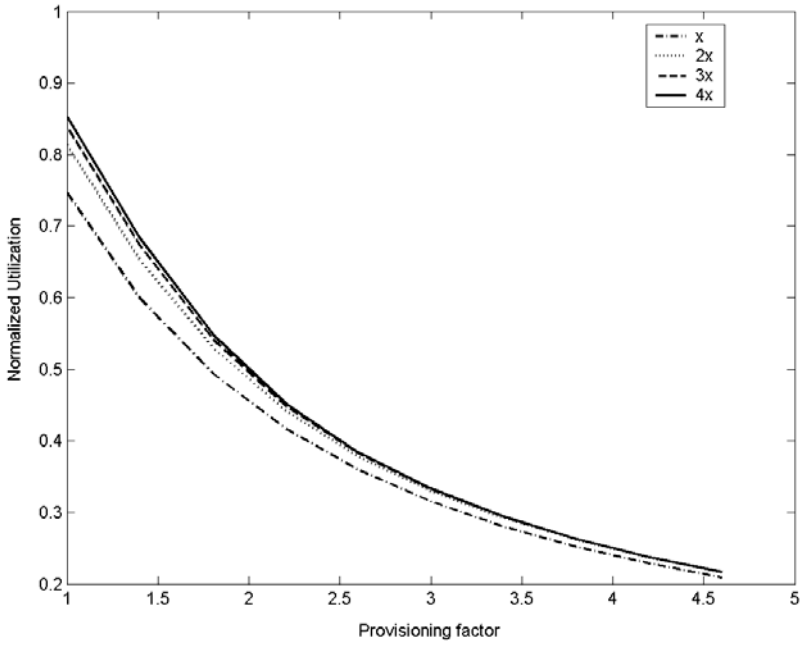


Figure 6.11 Utilization as a function of PF and MPB

For single class simulations, the utilization will be exactly the inverse of *PF* if zero loss was encountered. If loss is experienced, delay might improve due to shorter queue lengths, but the utilization is degraded again. For this reason the aim is to identify the effective range in the single class simulation. This is done to get *PF* values for loss values approaching zero. The *PF* values obtained for the various classes are then used along with the effective range of *MPB* values in the multi-class analysis. In this analysis, the *MPB* is expressed as the maximum number of bytes allowed in the queue per sampling interval of 6ms. The *MPB* values were logged, like the *PF* values, as a factor of the average data rate for the complete trace. Fine tuning of the *MPB* is done in the multi-class simulation where dynamic bandwidth sharing among traffic classes applies.

6.3.4 Statistical Results

The described simulation was repeated 50 times for each class to get a consistent statistical result. The results for *PF* and *MPB* are discussed separately in the sequel:

Provisioning Factor

To improve visual interpretation, the results for *PF* values of each class are presented in histograms.

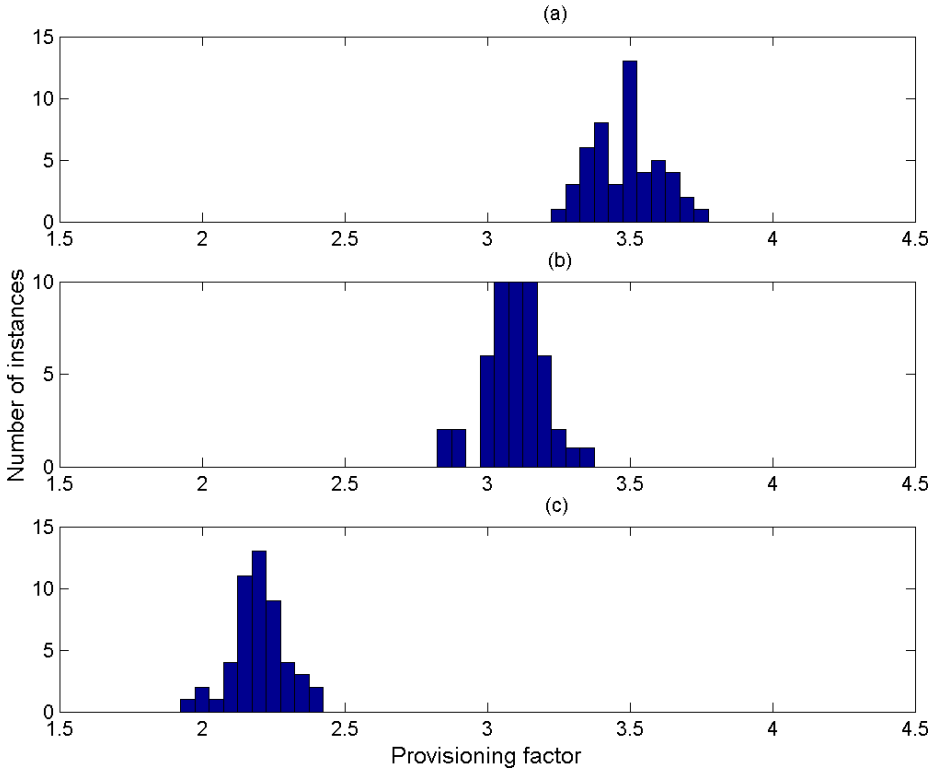


Figure 6.12 Histogram for PF values for the various classes

In figure 6.12, (a), (b) and (c) depict the *PF* for the RT, IB and BB classes respectively. The statistical properties of these values are given in table 6.1.

Table 6.1 Statistics of simulation results

	Minimum	Maximum	Mean	Variance	Std. deviation
RT	3.2537	3.7455	3.4820	0.0137	0.1168
IB	2.8635	3.3309	3.0970	0.0104	0.1022
BB	1.9551	2.4213	2.1948	0.0090	0.0948

From these results, the following conclusions can be made:

Relatively small variances confirm the use of these values. Considering the effect of statistical multiplexing under multi-class operation, such small variances become insignificant.

The generated traffic does not include Layer 1 and 2 overheads. If these *PF* and *MPB* values are used for real network dimensioning, the particular overhead percentages as specified in Chapter 3, should be added. Management traffic is included in the IB class. The IB *PF* should thus further be increased by another 10%.

Maximum Permitted Burst

The results obtained for *MPB* shows a rather scattered distribution without a clear relationship to the average data rate. No curve fitting was done for these results. However, an expression for the *MPB* in terms of the *Committed Information Rate* (CIR) was derived heuristically and compared it with simulation results. The heuristic argument was based on the purpose of the *MPB* including the following:

- To limit the delay experienced by its associated class.
- To prevent high classes to deprive lower classes from their fair share in the bandwidth.
- To limit losses.

The first two aspects tend to decrease the *MPB*, while the third aspect tends to decrease the *MPB*. In order to comply with the first aspect, the *MPB* can be computed mathematically. The server rate, *SR*, for a given class is given by

$$SR = PF \times CIR, \tag{6.1}$$

where *PF* is the provisioning factor and *CIR* is the committed information rate (bytes/s). Let *D* denote the permitted delay. The server can transmit a maximum of

$$D \times SR \ \ bytes \tag{6.2}$$

within the permitted delay period. Deriving *SR* in terms of *PF* and *CIR*, the *MPB* is given by

$$MPB = PF \times D \times CIR. \tag{6.3}$$

Throughout the simulations, values for *MPB* resulting in the required QoS, were logged from the graphs similar to the one depicted in figure 6.9. The *MPB* values are plotted as a function of the *CIR* in figures 6.13 to 6.15 for the RT, IB and BB classes respectively. In each case, the maximum and minimum factors that could possibly be obtained from the simulation results are indicated as well.

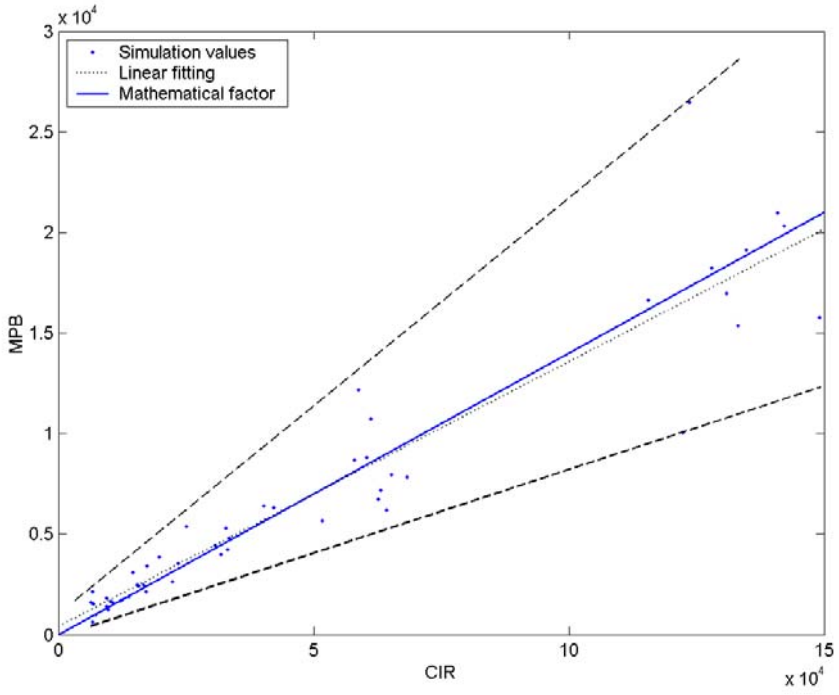


Figure 6.13 MPB vs. CIR for RT data

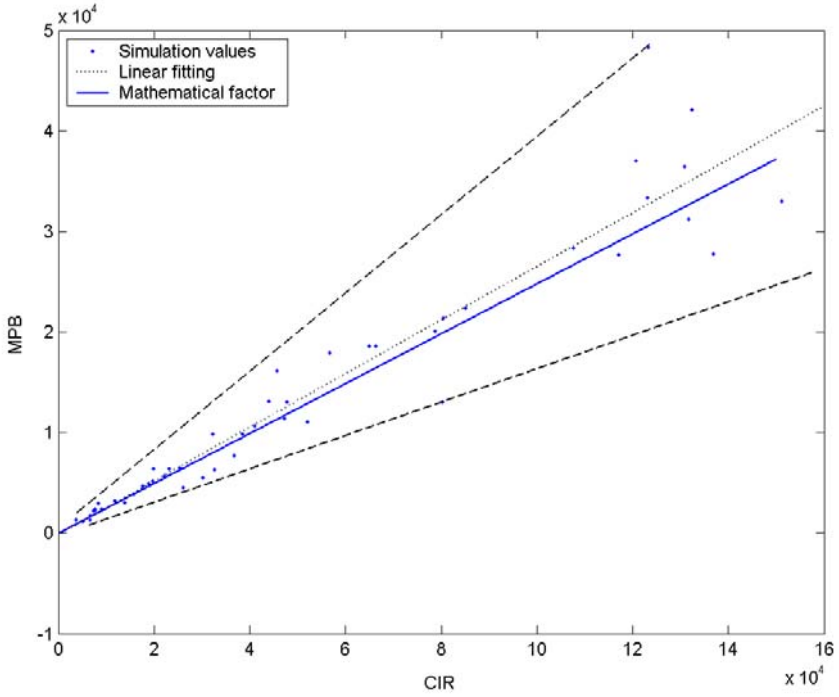


Figure 6.14 MPB vs. CIR for IB data

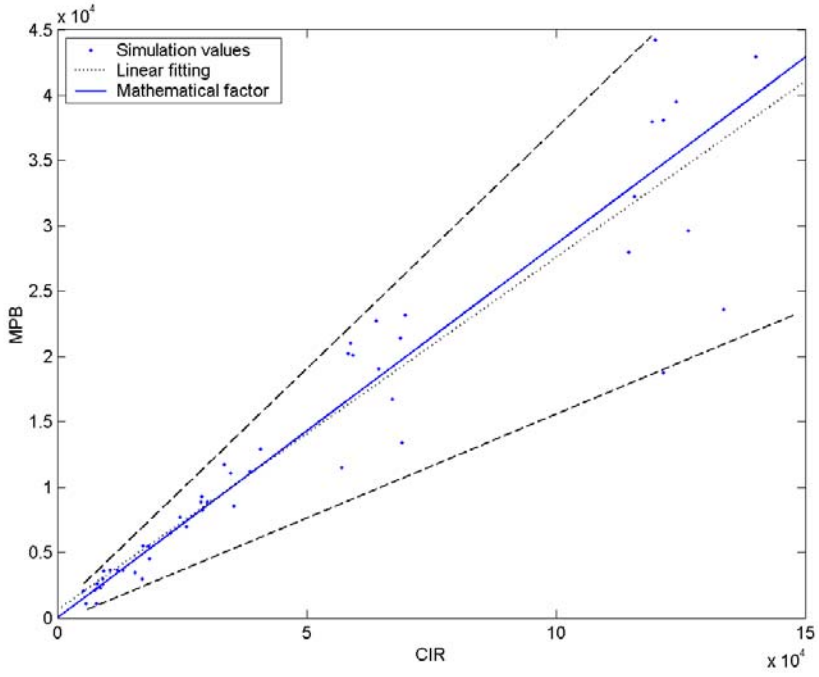


Figure 6.15 MPB vs. CIR for BB data

There is a correspondence between the simulation results and the mathematical values, but the variances from the mathematical values are large. Table 6.2 lists the minimum and maximum *MPB* values as well as the mathematical *MPB* value as a factor of the CIR.

Table 6.2 *MPB* values as a function of the CIR

	Minimum	Mathematic value	Maximum
RT	0.083	0.140	0.220
IB	0.166	0.248	0.390
BB	0.182	0.286	0.375

6.4 MULTI-CLASS SIMULATION

The simulation test bed described in section 5.6 was used for multi-class simulations. Tests were performed involving all the traffic classes. The *PF* values, as determined in the single class simulations, were used. The method described in Chapter 3 was used to compute the *combined server rate*, hereafter called the *CSR*.

Equations 6.3 and 6.4 were used to compute theoretical MPB values, for the RT class and the other classes respectively. Equation 6.3 results in a lower value than 6.4. This was chosen to prevent priority queue from depriving lower classes of their share of the bandwidth. For the other classes sharing the remaining bandwidth, the actual server rate that any class will experience while being served, is the *CSR*. For this reason the theoretical MPB for the fair queues, was determined by

$$MPB = CSR \times D. \tag{6.4}$$

Simulations were started with MPB. During each simulation, these values did not always produce the required QoS. They were adjusted to optimize the delay and loss according to the QoS parameters for each particular class. There was no need to adjust *SR*, only values were changed. These optimized MPB values are denoted as ${}^{*}MPB$. The ratio of ${}^{*}MPB$ and MPB, given by

$$\frac{^{*}MPB}{MPB}, \tag{6.5}$$

were logged throughout simulations. These ratios are depicted in figure 6.16.

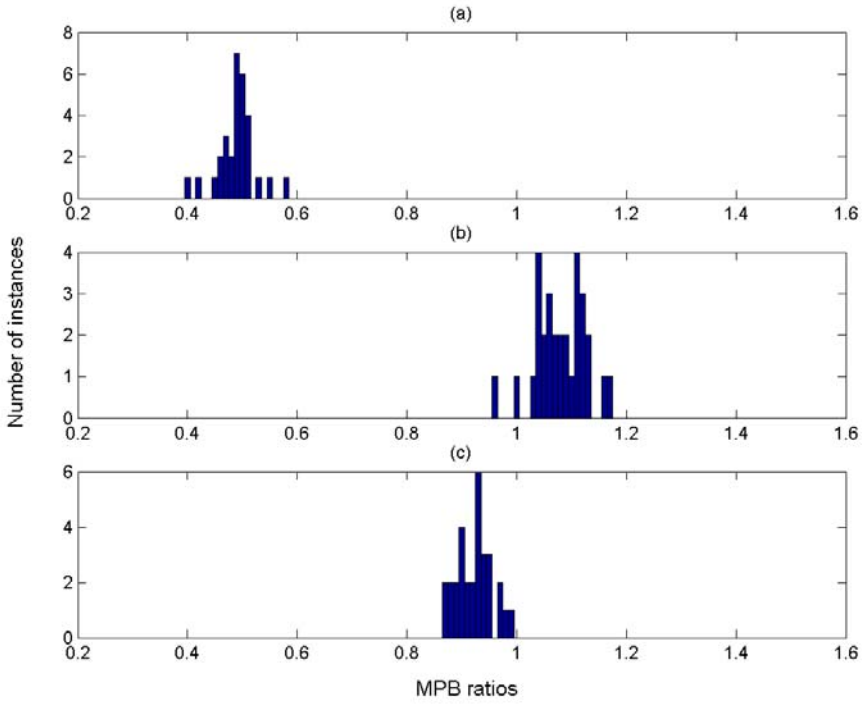


Figure 6.16 Histograms for MPB ratios for (a) RT, (b) IB and (c) BB data

The low ratio for the RT class can be contributed to the RT class's tight delay bound and tolerance for loss. The distribution around 1 for the IB and BB classes establish that the theoretically determined *MPB* serves as a good average in a multi-class environment as well. The BB class shows a lower average ratio than the IB class. This is due to the larger permitted delay which is already included by equation 6.4 for the BB class. With the optimized *MPB* values the following delay measurements were made for 20 multi-class simulations.

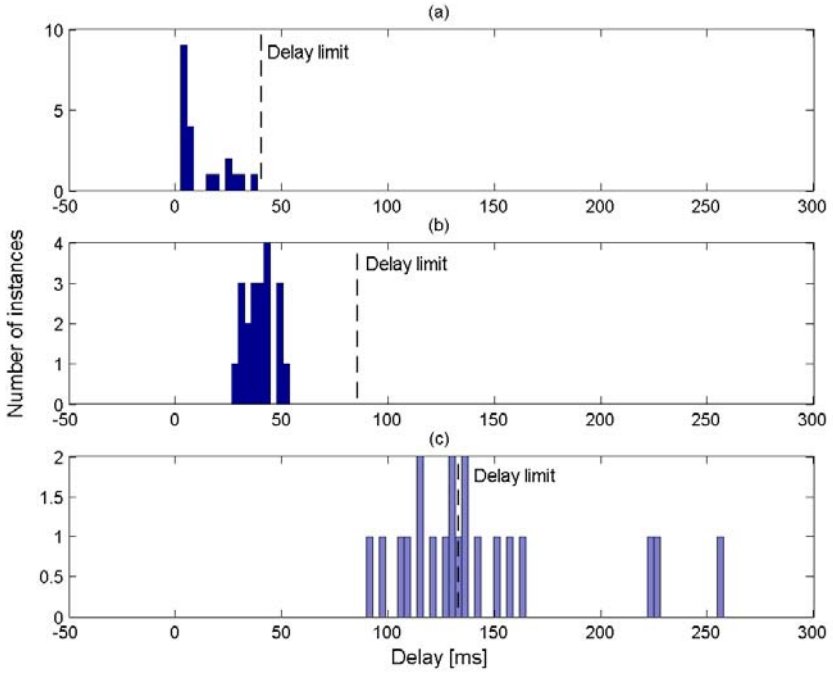


Figure 6.17 Histograms of delay measurements for (a) RT, (b) IB and (c) BB

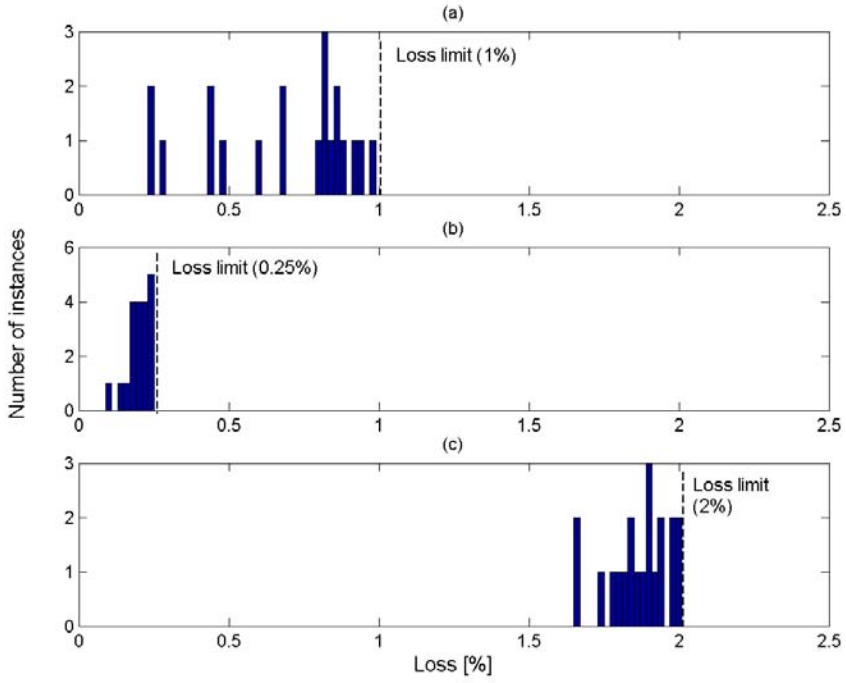


Figure 6.18 Histograms of loss measurements for (a) RT, (b) IB and (c) BB

The GD class is a best effort service without any guarantees. For this reason, complete results for the GD class are not presented. However, these results confirm the nature of a best effort service for the GD class. Throughout the simulations, GD revealed loss values between 0% and 6,7% depending on the *MPB* values. Delay measurements ranged between 0.78s and 10s. Many applications do not need the quality guarantees of the higher classes and are thus classified in the GD class. This class-based differentiation allows for excellent utilization values. Figure 6.19 shows total utilization values obtained throughout simulation, while providing the strict quality guarantees.

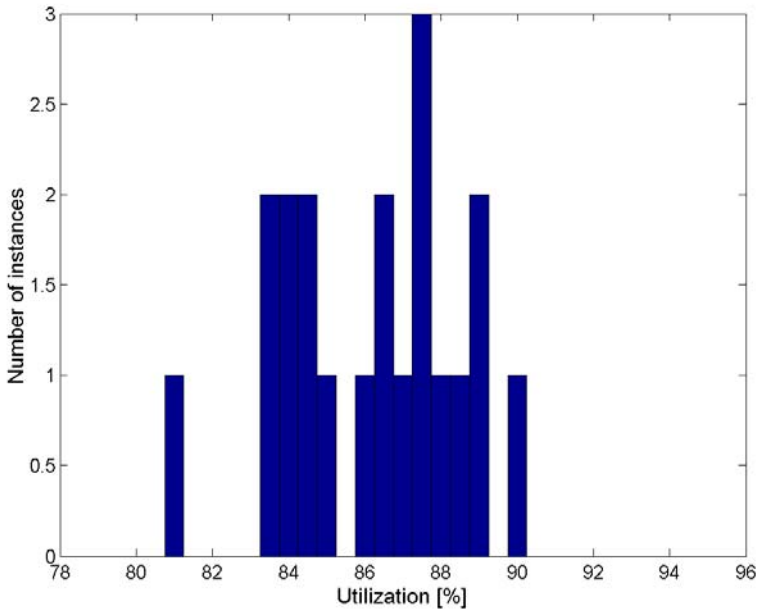


Figure 6.19 Histogram of utilization obtained on the single link for all the classes

6.5 DISCUSSION

PF results obtained from single-class simulation proved to be sufficient in a multi-class environment as well. Throughout simulations, all classes achieved the desired QoS. Only the BB class occasionally experienced higher delay than was specified. This can be seen from figure 6.17. It was found that this was not due to under-provisioning, since the RT and IB classes were performing exceptionally well during the same simulation. We conclude that two possible solutions exist. Firstly, the delay guarantee for the BB class could be made higher. If service providers believe that the specified delay parameter is needed, then the second solution

should be carried out. In this case the bandwidth distribution[12] among the fair queues could be adjusted.

The optimized *MPB* values vary for each trace. This can be attributed to the bursty nature of network traffic. However, the mathematical *MPB* values show a good average accounting for both delay and loss. For a fixed *PF*, there exists a tradeoff between delay and loss. This tradeoff can be controlled by the *MPB*. It is recommended that the relative importance of each QoS parameter should be considered to choose a *MPB* value between the specified minimum and maximum values. When delay is the most critical parameter, a lower *MPB* is recommended. If the loss specification is the most important, a higher *MPB* is recommended.

6.6 COMPARISON WITH EXISTING VALUES

6.6.1 Provisioning Factors

The provisioning factors, based on best practice recommendations, used for Telkom's traffic classes are compared to results obtained from this study in table 6.3.

Table 6.3 Comparison of PF values

	Telkom *PF*	Simulation *PF*	Simulation *PF* + overhead
RT	4	3.4820	3.89
IB	4	3.0970	3.81
BB	2	2.1948	2.48
GD	1	1.0000	1.00

Lower *PF* values were obtained for the RT and IB class and a higher value for the BB class. If a total package consists of equal demands for each class, approximately the same total line rate will be allocated. However if the demands for the various classes differ, the allocated bandwidth can differ by as much as 24%

[12] Currently, a distribution of 80% for IB, 15% for BB and 5% for GD are used. These values are as configured on routers throughout the DiffServ domain

6.6.2 Maximum Permitted Burst

Our results conclude that the use of equation 6.4 results in the acceptance of an average burst size accounting for both delay and loss specifications. We also recommend deviation from this value when either delay or loss is more important.

In current practice, the *MPB* values are not determined as a function of the *CIR*. It is computed theoretically by using the data rate defined for a session, the MTU, and the number of concurrent sessions per class. All these parameters are maximum values. Therefore, a worst case value is the result. Worst case results can generally not be compared to statistical results. However, simulation results give values of approximately a third of the theoretical worst case values.

6.7 CHAPTER SUMMARY

This Chapter started by describing the traces used for analysis. Traffic models were evaluated to be used for simulation purposes. Only the evaluation of results for the selected models to be used in simulations is presented. Results for the single class analysis described in Chapter 5 are presented. The values obtained from the single class simulation were used in the multi-class simulation to evaluate the effect of statistical multiplexing. Simulations resulted in satisfactory fixed results for provisioning factors (PF). Values for the maximum permitted burst (MPB) varied, but an equation is provided for determining a suitable MPB value, accounting for delay as well as loss. Deviations from this value, obtained through simulations, are also presented. The results obtained are shortly discussed and compared with values used in practice.

CHAPTER 7
SUMMARY AND CONCLUSION

7.1 SUMMARY

A prime concern of this dissertation is the increasing need for service differentiation and proper dimensioning in modern packet-switched networks. Service differentiation is based on the concept of Quality of Service (QoS). Chapter 2 introduces the concept of QoS, as the performance of a number of different network traffic measurements. The relation between QoS parameters and different types of applications is described. Various QoS mechanisms are discussed; with Differentiated Services (DiffServ) being the main focus of this dissertation.

The DiffServ domain, architecture, and functionality are described in Chapter 3. Packets traversing a DiffServ domain are aggregated in classes according to their applications. Each class receives a particular treatment across the domain. In this dissertation four classes are considered: Real Time (RT), Interactive Business (IB), Bulk Business (BB) and General Data (GD). The QoS parameters that describe these classes should be satisfied in practice, and that is set as one of the goals of the simulations.

Various network traffic characteristics are studied in Chapter 4 to support accurate traffic modeling. The concept of *self-similarity* is introduced and illustrated by means of visual examples and mathematical formulation. The concepts of *long range dependency* (LRD) and *short range dependency* (SRD) in network traffic are introduced and explained. It was observed that TCP traffic has a much higher LRD than UDP. This observation led to the use of two separate models for TCP and UDP respectively.

The chaotic way in which network resources are demanded, results in the *bursty* nature of network traffic. It was observed that maximum utilization is exponentially inversely proportional to the burstiness of an arrival process. Empirical data shows that network traffic, especially UDP is very bursty. From this observation it is concluded that optimal utilization could be achieved if mechanisms are developed to rearrange application traffic into a smoother data stream. This implies an arrival process with a low burstiness resulting in good utilization.

Methods to measure and quantify the self-similarity, LRD and burstiness are introduced. A number of SRD and LRD traffic models are described for possible use in simulations.

Chapter 5 is concerned with theoretical considerations and evaluation criteria regarding traffic traces and traffic models to be used in simulations. The procedures for traffic generation and the setup for simulation test beds are described for single as well as multi-class analysis. The particular QoS parameters required for each class are introduced. Lower layer overheads are investigated, since dimensioning is done only at the Network layer.

In chapter 6 the main simulation results are presented. Firstly, results for evaluation of the traffic models are given. Dimensioning values were obtained from single class simulation results. The multi-class analysis evaluated the *provisioning factors*, obtained from the single class simulation under the effect of statistical multiplexing. Results for the multi-class simulation confirmed that these values are appropriate. A comparison between our experimental results and the values that are used in practice show that the RT and IB classes require less bandwidth, while the BB class should be allocated more bandwidth.

7.2 CONCLUSION

Proper resource dimensioning requires knowledge of various fields of study. This dissertation contributes to the question of resource dimensioning by surveying these fields in a coherent way. Further, more suitable dimensioning values for four distinct traffic classes in a mixed traffic environment were derived. These values were obtained by analyzing the dynamic

behavior of various QoS parameters under realistic traffic load conditions, by means of computer simulations.

DiffServ is a relatively new architecture, and the values that are currently being used in dimensioning models are based on laboratory tests, intuitive guessing, or reference to best practice recommendations. This dissertation aims to determine dimensioning values by means of simulations. It enhances traditional simulation methods by using two separate traffic models for UDP and TCP respectively.

The statistical nature of an arrival process greatly affects the performance of a network. Therefore traffic models that are used in simulations need to be very accurate to capture the statistical properties of real network traffic. Queuing behavior, correlation matching and marginal distributions were used to evaluate a number of traffic models. Results show that the Multi-fractal Wavelet Model (MWM) [44], [45] and the Markov Modulated Poisson Process (MMPP) [66] are best suited to capture TCP and UDP characteristics respectively.

Simulations were performed on single and multi-class traffic, and both gave satisfactory results. The single class simulation resulted in provisioning factors with a low variance for each class. Results obtained for the *maximum permitted burst* (MPB) values were scattered with very huge variances. This can be attributed to the bursty nature of network traffic. Theoretical MPB values were derived for each class. Larger values than the theoretical MPB, resulted in a lower loss, but required a higher delay. Likewise, smaller MPB values resulted in better delay measures, but led to increased loss. It is recommended that the relative importance of the delay and loss parameters within each class should contribute to the establishment of the MPB value.

Multi-class simulations confirmed that the values obtained from the single class perform well under the effect of statistical multiplexing. Occasional simulation results where the delay exceeds the specified value for the BB class can be attributed to the bandwidth allocation among the classes that do not correspond to delay specifications. In practice, distributions of 80%, 15% and 5% are used for the IB, BB and GD classes respectively. Based on results it is

recommended that these values should be changed, or alternatively that the delay guaranteed for the BB class should be increased.

These results can be used separately or in conjunction with existing values when dimensioning tasks are being performed. This study establishes how dimensioning values obtained from laboratory experiments and recommendations correspond to values obtained from simulation results. Results are further leading to recommendations that will assist service providers to deliver services more accurately according to the Service Level Agreement (SLA).

7.3 FUTURE WORK

All the objectives stated in Chapter 1 were achieved. One way in which this work could be extended would be to implement the traffic models on a network simulator, such as AdTech described in Chapter 5, running over a live network. In this way, a more realistic network environment would be obtained, possibly giving more accurate results.

REFERENCES

[1] P. Simpson, "Out with the old", *Telecommunications Magazine*, June 2004, Available at: http://www.telecommagazine.com/default.asp?journalid=3&func=articles&page=0406t13&year=2004&month=6. Last visited on 1 October 2004.

[2] T. H. Nguyen and M. N. O. Sadiku, "Next generation networks", *Potentials, IEEE,* vol. 21, issue 2, April-May 2002, pp. 6-8.

[3] M. Johnson and D. Ludlam, "Evaluation of alternative service delivery mechanisms for next generation networks", *Intelligent Network Workshop, IEEE,* May 2001, pp 325 – 335.

[4] D. Goderis, S. Van den Bosch, Y. T'Joens, P. Georgatsos, D. Griffin, G. Pavlou, P. Trimintzios, G. Memenios, E. Mykoniati, C. Jacquenet, "A service-centric IP quality of service architecture for next generation networks", *Network Operations and Management Symposium, IEEE/IFIP*, 15-19 April 2002, pp.139 – 154.

[5] J. van Greunen and R. Achterberg , "A Network Operator's perspective on IP VPNs", presented at the Southern African Telecommunication Networks and Applications Conference (SATNAC) conference, George, September 2003.

[6] Cisco Systems, "DiffServ-The scalable End-to-End Quality of Service Model", 2001. Available at: http://www.itpapers.com/search.aspx?scid=45&x=40. Last visited on 1 October 2004.

[7] C. Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines", Juniper Networks, 2001. Available at: www.juniper.net/solutions/literature/white_papers/200020.pdf. Last visited on 1 October 2004.

[8] ITU-T Recommendation G.114, "One Way Transmission Time", 2002.

[9] Cisco Systems, "Understanding Delay in Packet Voice Networks", 2004. Available at: http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml. Last visited on 1 October 2004.

[10] V. Firoiu, M. Borden, "A study of active queue management for congestion control", *INFOCOM 2000. Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 26-30 March 2000, pp.1435 – 1444.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", Network Working Group, RFC2475, 1998.

[12] Information Sciences Institute, University of Southern California, "Internet Protocol", RFC 791, 1981.

[13] K. Nichols, S. Blake, F. Baker, D. Blacket, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", Network Working Group, RFC2474, 1998.

[14] ITU-T I Series Recommendations, 2003. Available at: http://www.itu.int/publications/itu-t/index.html. Last visited on 1 October 2004.

[15] IEC, "Trends in the deployment of SDH", 2004. Available at: http://www.iec.org/online/tutorials/sdh/topic07.html. Last visited on 1 October 2004.

[16] S. Brim, B. Carpenter, F. Le Faucheur, "Per Hop Behavior Identification Codes", Network Working Group, RFC2836, May 2000.

[17] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", Network Working Group, RFC2597, June 1999.

[18] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", Network Working Group, RFC2598, June 1999.

[19] R. J. Majoor, "Quality of Service in the Internet Age, "Proceedings of the Southern African Telecoms Networks and Applications Conference (SATNAC) conference, September 2003.

[20] C. Finseth, "An Access Control Protocol, Sometimes Called TACACS", Network Working Group, RFC1492, July 1993.

[21] R. Ullmann, "TP/IX: The Next Internet", Network Working Group, RFC1475, June 1993.

[22] Cisco Systems, "Configuring Frame Relay Traffic Shaping", 2003. Available at: http://www.cisco.com/en/US/tech/tk713/tk237/technologies_configuration_example09186a00800942f8.shtml. Last visited on 2 October 2004.

[23] Cisco Systems, "Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting", 2003. Available at: http://www.cisco.com/en/US/tech/tk543/tk545/technologies_tech_note09186a00800a3a25.shtml. Last visited on 2 October 2004.

[24] R. J. Majoor, "A MAC Protocol for Wireless Networks with QoS Guarantees", Ph.D. Dissertation, Dept. Elect. Eng., University of Natal, South Africa, 2002.

[25] S. Bhatti and J. Crowcroft, "QoS-sensitive flows", *IEEE Internet computing*, vol. 4, issue 4, July/August 2000, pp. 48-57.

[26] S. Jung, J. Kwak, O. Byeon, "Performance analysis of queue scheduling mechanisms for EF PHB and AF PHB in DiffServ networks", *High Speed Networks and Multimedia Communications 5th IEEE International Conference*, 3-5 July 2002, pp.101-104.

[27] C. Shan, "A new scalable and efficient packet scheduling method in high-speed packet switch networks", *High Performance Switching and Routing, 2001 IEEE Workshop*, 29-31 May 2001, pp.16 – 20.

[28] L. Jinhui, N. Ansari, "QoS guaranteed input queued scheduling algorithms with low delay", *High Performance Switching and Routing, IEEE Workshop*, 29-31 May 2001, pp. 412 – 414.

[29] R. L. Cruz, "A Calculus for Network Delay: Parts I and II", *IEEE transactions on Information Theory*, vol. 37, issue 1, January 1991, pp. 114-141.

[30] T Bonald, A. Proutiere, J. Robberts, "Statistical Performance Guarantees for Streaming flows using Expedited Forwarding", *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Infocom*, vol. 2, 22-26 April 2001, pp 1104 - 1112.

[31] K. Kumaran and M. Mandjes, "Multiplexing Regulated Traffic streams: Design and Performance", *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE infocom*, vol. 1, 22-26 April 2001, pp 527 – 536.

[32] S. Rajagopal, M. Reislein, K. Ross, "Packet Multiplexers with Adversarial Regulated Traffic", *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, 29 March - 2 April 1998, pp. 347-355.

[33] M. Reislein, K. Ross, S. Raijagopal, "A framework for Guaranteeing Statistical QoS", IEEE/ACM Transactions on Networking, vol. 10, issue 1, February 2002 pp. 27 – 42.

[34] Cisco Systems, "Low Latency Queuing", 2004. Available at http://www.cisco.com/en/

US/tech/tk543/tk544/tk399/tech_protocol_home.html. Last visited on 2 October 2004.

[35] S. Taha and M. Kavehrad, "Dynamic bandwidth allocation in multi-class connection-oriented networks", *Computer Communications, In Press, July 2003, Corrected Proof*. Available online at www.ComputerScienceWeb.com. Last visited on 2 October 2004.

[36] M. López-Guerrero, J. Gallardo, L. Orozco-Barbosa and D. Makrakis, "On the dynamic allocation of resources using linear prediction of aggregate network traffic", *Computer Communications*, vol. 26, issue 12, July 2003, pp. 1341-1352.

[37] S. Low, P.Varaiya, "An algorithm for optimal service provisioning using resource pricing", *INFOCOM '94, Networking for Global Communications. 13th Proceedings IEEE,* vol 1, 2-16 June 1994, pp. 368 -373.

[38] S. Sato, K. Kobayashi, H. Pan, S. Tartarelli, A. Banchs, "Configuration Rule and Performance Evaluation for DiffServ Parameters", *Teletraffic Engineering in the Internet Era*, Elsevier, vol.17, Issue 4, 2001, pp 931-942.

[39] J. Hwang, H. Kim and M. Weiss, "Interprovider differentiated service interconnection management models in the Internet bandwidth commodity markets", *Telematics and Informatics*, vol. 19, issue 4, Nov. 2002, pp. 351-369.

[40] T. Anjali, C. Scoglio and G. Uhl, "A new scheme for traffic estimation and resource allocation for bandwidth brokers", *Computer Networks*, vol. 41, issue 6, April 2003, pp. 761-777.

[41] Q. Jing-yu and E. Knightly, "Inter-Class Resource Sharing using Statistical Service Envelopes", *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol. 3, 21-25 March 1999, pp 1404 – 1411.

[42] V. Paxson, S. Floyd, "Wide area traffic: the failure of Poisson modelling", *Transactions on Networking, IEEE/ACM*, vol. 3, issue 3, June 1995, pp. 226-244.

[43] W. Shengquan, " Providing absolute differentiated services with statistical guarantees in static-priority scheduling networks", *Proc. Seventh IEEE Real-Time Technology and Applications Symposium*, 30 May-1 June 2001, pp. 127 -129.

[44] A. du Plessis and M. A. van Wyk, "Modelling of Ethernet Traffic with self-similar processes", presented at the Southern African Telecommunication Networks and Applications Conference (SATNAC) conference, George, September 2003.

[45] R.H. Riedi, M.S. Crouse, V.J. Ribeiro, R.G. Baraniuk, "A Multifractal Wavelet Model with Application to Network Traffic," *IEEE Transactions on Information Theory*, vol. 45, no. 3, 1999, pp. 992-1018.

[46] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, 1995, pp. 953-962.

[47] J. W. Roberts, "Traffic theory and the internet", *IEEE Communications Magazine,* vol. 39, issue 1, January 2001, pp 94-99.

[48] Z. Hulicki, "On some aspects of modelling and dimensioning in broadband access networks", *Proceedings of the 31st Annual Simulation Symposium,*5-9 April 1998, pp. 106 -109.

[49] A. Adas, "Traffic models in broadband networks", *Communications Magazine, IEEE ,* vol. 35, issue 7, July 1997, pp. 82 -89

[50] W. Stallings, High speed networks and Internets: Performance and quality of service, 2nd ed. Prentice Hall, 2001.

[51] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the Self-Similar Nature of Ethernet Traffic," *IEEE/ACM Trans. Networking*, vol. 2, issue 1, February 1994, pp. 1-15.

[52] J. Beran, R. Sherman, M. S. Taqqu, W. Willinger, "Variable-Bit-Rate Video Traffic and Long-Range Dependence", accepted for publication in *IEEE Trans. On Communication*, subject to revisions, 1995.

[53] D. R. Cox, "Long-Range Dependence: A Review", *Statistics: An Appraisal*, *The Iowa State University Press, Ames, Iowa*, 1984, pp. 55-74.

[54] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian Motions, Fractional Noises and Applications", *SIAM Review 10*, 422-437, 1968.

[55] D. L. Jagerman, B. Melamed, and W. Willinger, "Stochastic modelling of traffic processes*," Frontiers of Queuing: Models, Methods and Problems, CRC Press*, 1997, pp. 271-370.

[56] C. Fraleigh et al, "Packet-level traffic measurements from the sprint IP backbone", *Network, IEEE* , vol.17, Issue 6, Nov.-Dec. 2003, pp. 6 -16.

[57] Sprint IPMON DMS, "Protocol Breakdown", 2003. Available at http://ipmon.sprint.com/packstat/viewresult.php?1:protobreakdown:nyc-21.0-030407. Last visited on 2 October 2004.

[58] T. Kushida, "An empirical study of the characteristics of Internet traffic", *Computer Communications*, vol. 22, issue 17, 15 October 1999, pp 1607-1618.

[59] Novel Articles, "The Impact of Packet Size Distributions", 1996. Available at http://developer.novell.com/research/appnotes/1996/january/03/01.htm. Last visited on 2 October 2004.

[60] Engineering Statistics, Arizona State University, John Wiley & Sons, Inc, 1998, pp. 131 – 209.

[61] M.W. Garret and W. Willinger, "Analysis, modelling and generation of self-similar VBR video traffic", *Proceedings ACM SIGCOM '94*, London 1994, pp. 269-279.

[62] J. Cosmas, S. Manthorpe, A. Odinma-Okafor, R. Grunenfelder, "Characterisation of variable rate video codecs as autoregressive moving average processes for ATM networks", *Proceedings of the third IEE Conference on Telecommunications*, 17-20 March 1991, pp. 353 –359.

[63] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, A. Odinma-Okafor, "Characterization of Video Codecs as Autoregressive Moving Average Processes and Related Queuing System Performance", *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, 1991, pp. 284-293.

[64] W. Vandaele, *Applied Time Series and Box-Jenkins Models*, Florida: Academic Press Inc., ISBN 0-127-12650-3, 1983, pp. 32-60.

[65] Y. Kim, S.Q. Li, "Time-scale of interest in traffic management for link bandwidth allocation design", *Proc. IEEE Infocom'96 Conference*, 1996, pp. 738–748.

[66] H. Heffes and D.M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance", *IEEE Journal on Selected Areas in Communications*, vol.4, 1986, pp.856-868.

[67] B. Maglaris, D. Anastasiou, P. Sen, G. Karlsson, J.D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Transactions on Communications*, vol. 36, no. 7, 1988, pp. 834-844.

[68] B. Melamed and B. Sengupta, "TES modelling of video traffic", *IEICE Transactions on Communications*, vol. E75-B, no.12, 1992, pp. 1292-1300.

[69] Reichl, P., "A generalized TES model for periodical traffic", *Proceedings of the Int. Conf. on Computer Communications (ICC'98)*, *Atlanta (GA),* vol. 3, 7-11 June 1998 pp. 1461 - 1465.

[70] B. Ryu, A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities", ACM Computer Communications. Rev. 26 (1996), pp. 3–14.

[71] K. Nagarajan, "Fractional ARIMA Processes and Its Applications in Network Traffic Modelling", Ph.D dissertation., Center for Signal and Image Processing, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1998.

[72] G. Gripenberg & I. Norros, "On the Prediction of Fractional Brownian Motion", *Journal of Applied Probability*, vol. 33, 1996, pp. 400-410.

[73] V. Paxson, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic", *SIGCOMM Computer Communication Review*, vol. 27, no. 5, 1997, pp. 5-18.

[74] C.S. Burrus, R.A. Gopinath & H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*, New Jersey: Prentice Hall, ISBN 0-13-489600-9, 1998, pp.1-30.

[75] J.B. Kim, R. Simha, and T. Suda, "Analysis of a Finite Buffer Queue with Heterogeneous Markov Modulated Arrival Processes: A Study of Traffic Burstiness and Priority Packet Discarding", *Computer Networks and ISDN Systems*, vol. 28, no. 5, 1996, pp. 653-673.

[76] D. Heyman, T. Lakshman, "What are the implications of long-range dependence for VBR-video traffic engineering", *IEEE Trans. Network,* vol. 4, issue 3, 1996.

[77] N. Likhanov, R. Mazumdar, "Cell loss asymptotics in buffers fed with a large number of independent stationary sources", *Seventeenth Annual Joint Conference of the IEEE*

Computer and Communications Societies. Proceedings. IEEE INFOCOM , vol. 1, 29 March - 2 April 1998, pp. 339 – 346.

[78] A. Sang and S. Li, "A predictability analysis of network traffic", *Computer Networks*, vol. 39, issue 4, 15 July 2002, pp 329-345.

[79] Y. T Hou,.T.; Z. Duan; Z. Zhang; T. Chujo, "Providing scalable support for multiple QoS guarantees: architecture and mechanisms", *Communications, ICC, IEEE International Conference,* 11-14 June 2001, vol. 7, pp 2115 -2122.

[80] T. S. Randhawa, R.H.S. Hardy, "Performance evaluation of bandwidth partitioning in broadband networks", *High Performance Switching and Routing, Proceedings of the IEEE Conference on ATM*, 26-29 June 2000, pp 411 -418.

[81] G. Ghinea, G. A. Magoulas, C. Siamitros, "Perceptual, considerations in a QoS framework: a fuzzy logic formulation", *Fourth Workshop on Multimedia Signal Processing, IEEE*, 3-5 Oct. 2001, pp 353 –358.

[82] M. Crovella, A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes", *IEEE Transactions on Networking,* December 1996, vol. 5, issue 6, pp. 835-846.

[83] Adtech AX4000 Network Simulator reference page, Spirent Communications. Available at: http://www.tine.nl/marketplace/mypage/products_detail.asp?mypageid=459&productid=452. Last visited on 2 October 2004.

[84] Cisco Systems, "Measuring the Utilization of ATM PVCs", 2002. Available at: http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080093c9a.shtml. Last visited on 2 October 2004.

[85] J. van Greunen, V. Naidoo, H. Morar, "Dimensioning and provisioning links for IP quality of service", unpublished.

[86] LBL-PKT traces. Available at: http://ita.ee.lbl.gov/html/contrib/LBL-PKT.html. Last visited on 2 October 2004.

APPENDIX A
STATISTICAL DEFINITIONS

This chapter briefly describes the statistical definitions needed for the study of network traffic analysis and modelling.

A.1 TIME SERIES

Time series models are commonly used for network traffic modelling, due to the need to model changes in the traffic over time.

Definition *A **random variable**, X, is a measurable function from a probability sampling space Ω, to some measurable (real number) space.*

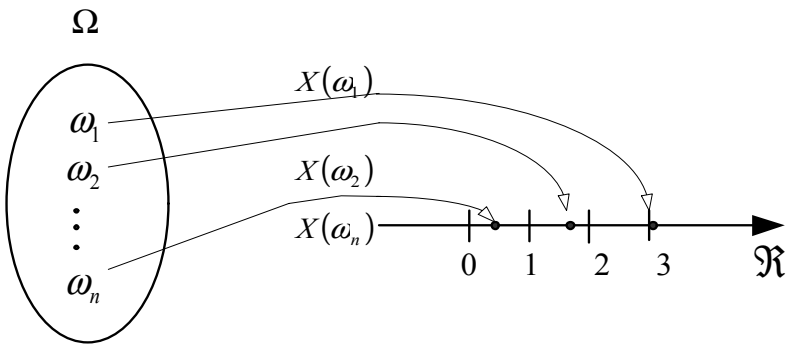


Figure A.1 Random variable

Definition A ***stochastic process***, X_t, is a sequence of random variables over time.

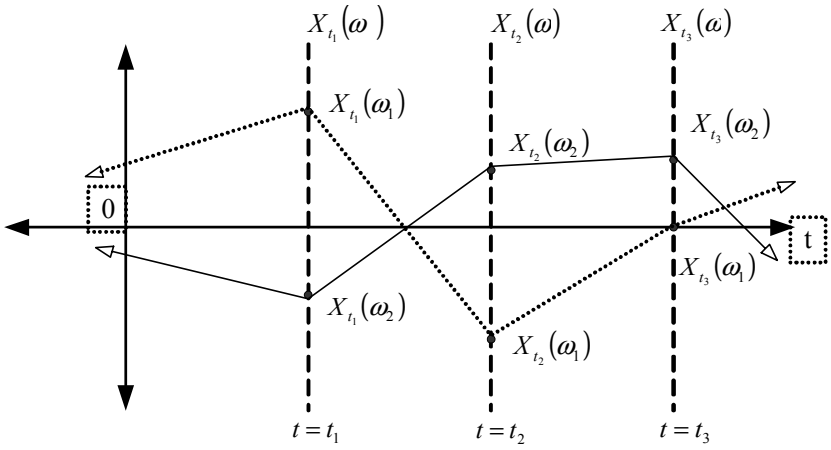


Figure A.2 A stochastic process

Note that for a fixed time t, $X_t(\omega)$ is a random variable and for a fixed sample ω, $X_t(\omega)$ is a function of t (or a signal) called a realization of ω. For all ω the functions $X_t(\omega)$ of t is called the ensemble.

Definition A ***time series*** is a single realization of a stochastic or deterministic process with a fixed ω. In general, a time series is denoted by

$$\{X(t) : t \in T\},$$

where $X(t)$ is the observation at time t, and T is the set of times at which observations were made. If the time between observations is equal then the notation will simplify to

$$\{X_t : t \in \mathbf{Z}\}.$$

If a mathematical model can exactly describe a time series it is called a *deterministic model*. In most cases the time series is *stochastic* because future values can only be partially accounted for by historic values.

Sampling the incoming traffic of a network will result in a time series where the samples have certain dependencies. Because of these dependencies, one can predict future values of the network traffic using the historic sampled values.

A.2 MEAN VALUE FUNCTION

The first moment of a stochastic process $\{X_t\}$ is only its expected value. The first moment is also known as the *mean* of the process. The processes sample mean can be approximated as

$$\mu(t) = E(X_t) \approx \frac{\sum_{t=1}^{n} X_t}{n},$$

where n is the number of samples in the stochastic process.

A.3 AUTO-COVARIANCE FUNCTION

The second mixed *central* moment of a stochastic process is known as the *auto-covariance* of the process and is calculated by

$$\gamma(t_1, t_2) = Cov(X_{t_1}, X_{t_2}) = E\{[X_{t_1} - \mu(t_1)][X_{t_2} - \mu(t_2)]\}.$$

The auto-covariance measures the similarity between two points on the same series observed at different times. The following notation for the auto-covariance will further be used

$$\gamma_k = \gamma(t, t+k).$$

The latter notation is justified for wide-sense stationary processes only.

A.4 VARIANCE FUNCTION

The second central moment of a stochastic process is known as the *variance* of the process and is calculated by

$$\sigma^2(t) = Var(X_t) = E\{[X_t - \mu(t)]^2\} = \gamma(t, t).$$

The variance measures the degree to which the process is spread out over the real line.

A.4 AUTOCORRELATION FUNCTION

The normalized *autocorrelation function* (ACF), ρ, of a data series, $X_t = \{X_1, X_2, \ldots\}$, is calculated as follows

$$\rho(t_1,t_2) = Cor(X_{t_1}, X_{t_2}) = \frac{Cov(X_{t_1}, X_{t_2})}{\sqrt{Var(X_{t_1})Var(X_{t_2})}} = \frac{\gamma(t_1,t_2)}{\sigma(t_1)\sigma(t_2)}.$$

The autocorrelation measures the linear predictability of the series at time t_1 using only the value X_{t_2}.

APPENDIX B

CLOSED-FORM WAVELET AND SCALING COEFFICIENT EXPRESSIONS

An indexing scheme is defined that indexes the possible shifts of the descendants of $U_{0,0}$ at scale j. This indexing scheme is used to obtain an expression for the ratio of variances in terms of the multipliers for the multi-fractal wavelet model.

Let, k_j, $j>0$, be the index of the possible shifts at scale j. The index of a descendant scaling coefficient is related to the parent scaling coefficient in the following manner: $k_{j+1} = 2k_j + k_j^{'}$, where $k_j^{'} = 0$ corresponds to the left descendant and $k_j^{'} = 1$ to the right descendant (see Figure B.1 below).

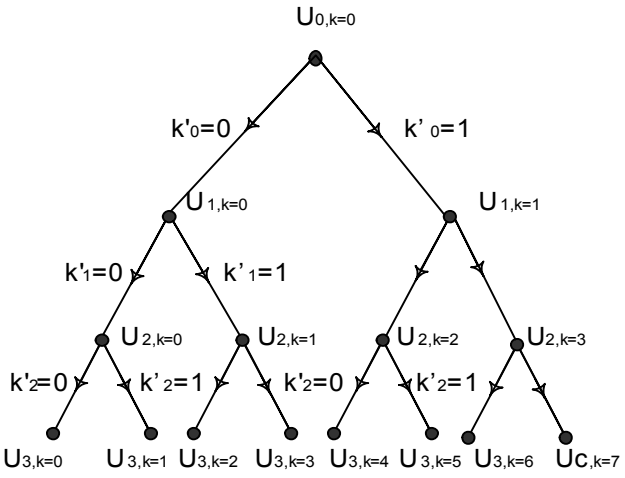


Figure B.1 Binary tree of the indexing scheme

We can express k_j as a binary expansion in terms of $k_j^{'}$ in the following manner

$$k_k = \sum_{i=0}^{j-1} k_j^{'} 2^{j-1-i}$$

By fixing a sequence of $k_j^{'}(i = 0,..., j)$ a line of descendants from $U_{0,0}$ down to U_{j,k_j} can be specified. For example, the sequence $k^{'} = \{1,0,0\}$ will lead to the scaling coefficient $U_{3,4}$ with the sequence of its parents, $\{U_{1,1}, U_{2,2}\}$.

It is important to note that $k_j = \left[\frac{k_{j+1}}{2}\right]$ and $k_j^{'} = k_{j+1} - 2\left[\frac{k_{j+1}}{2}\right]$ with $[x]$ the largest integer less than or equal to x