Measurement based Quality of Service Control for Communications Networks

Alan Davy, BSc

Department of Computing, Mathematics and Physics

Waterford Institute of Technology

Thesis submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

Supervisors : Dr Brendan Jennings and Dr Dmitri Botvich

September 2008

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Signed:.. ID: 19994248
Date: September 2008

Acknowledgements

I would first like to thank my supervisors Dr. Brendan Jennings and Dr. Dmitri Botvich, their excellent guidance and admirable patience provided me with the necessary support to see this thesis through to completion. They offered unrestricted access to their in-depth knowledge and experience from which I learned a great deal.

I would also like to thank the broader research support available to me at Waterford Institute of Technology, especially Mícheál Ó Foghlú, Dr. Willie Donnelly and Eamonn de Leaster who encouraged an atmosphere of research in WIT that inspired me to pursue and complete this task. I would like to thank Dr. Sven van der Meer and Dr. Sasitharan Balasubramaniam who continually encouraged me to pursue and progress my research along this path.

To my brother Steven how supported me through this process, our long conversations stimulated thought and invoked critical appraisal of the work within the thesis, which helped strengthen the fundamental research carried out. I would like to thank my loving parents Cormac and Irene. Their endearing encouragement carried me through the harder times, and ensured me there was always someone to turn to for support. To my brothers Robert, Brian and Wayne who were always there for me when I needed to unwind and relax. And finally to my friends and fellow band mates, who offered a much appreciated alternative to long and late nights of research and study.

Measurement based Quality of Service Control for Communications Networks

Alan Davy, BSc.

Supervisors: Dr. Brendan Jennings and Dr. Dmitri Botvich

Abstract

This thesis presents a purely empirical approach for the estimation of effective bandwidth of aggregated traffic flows, suitably accurate for use within traffic engineering based network performance optimisation strategies, such as QoS-aware network planning and admission control. The effective bandwidth estimation algorithm proposed, is an off-line approach, which uses a packet trace collected from the network and processes it through a FIFO queue. Based on queue buffer analysis, an appropriate queue service rate for the FIFO queue that ensures stated QoS targets on packet delay are maintained is chosen to represent the effective bandwidth estimation of the packet trace. The algorithm is evaluated in two traffic engineering based scenarios. The first is a cost efficient process for establishing a network-wide demand matrix from available network accounting data. We use effective bandwidth coefficients to enhance such a demand matrix for QoS-aware network planning. This process is demonstrated to be reasonably accurate for the scenarios we consider, and importantly at a fraction of the operational and capital expenditure that would be incurred by the deployment and operation of a direct measurement approach. Secondly, we propose two empirical effective bandwidth estimation based IPTV focussed admission control algorithms suitable for optimising utilisation of bandwidth while maintaining QoS targets on packet delay of admitted traffic. The first algorithm employs a simple evaluation of whether there is sufficient bandwidth available to ensure, with an appropriate degree of confidence, that QoS targets will not be violated if a requested flow is admitted. The second algorithm utilises information within the admission control process relating to the cost, duration and request frequency of specific IPTV content to prioritise flows that maximise revenue.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Quality of Service (QoS) control of traffic within a communications network is crucial for successful generation of revenue for the network operator. The network operator wishes to guarantee QoS of traffic while optimising the utilisation of network resources. Central to a network performance optimisation strategy for QoS control is the estimation of effective bandwidth, which is the minimum amount of bandwidth required on a link to maintain QoS targets of a traffic flow (Kelly, 1996). However the estimation of effective bandwidth is a non trivial task. Traditional approaches that are reliant on traffic model assumptions to estimate this value, can lead to sub-optimal utilisation of bandwidth resources, if used for performance optimisation. This thesis presents a purely empirical approach to effective bandwidth estimation that is suitable for the accurate estimation of minimum bandwidth requirements of a traffic flow, to maintain set QoS targets, independent of traffic model assumptions.

Current communication networks offer a variety of services to the customers including Video on Demand (VoD), Voice over IP (VoIP), IP Television (IPTV), as well as data services such as HTTP, email and FTP. Offering these services in compliance with contractual Service Level Agreements (SLA) to customers, is the main driver of revenue generation for the network operator. To ensure the QoS of traffic is provisioned correctly within the network while optimising bandwidth utilisation, the operator employs various traffic engineering strategies, all dependent on the accurate estimations of effective bandwidth. However it is accepted that effective bandwidth estimation approaches relying only on strict traffic model assumptions often lead to sub-optimal bandwidth utilisation and/or QoS control of traffic.

The following are a set of requirements the network operator must adhere to when developing performance optimisation strategies for QoS control within the network:

- **Maintain SLAs and QoS:** The network operator has a responsibility to its customers to maintain contractual SLAs in the provisioning of any services over its network. The reason for this is that customers are the main source of revenue for the network operator and failure to meet SLAs can result in loss of revenue. The network operator must ensure that any methods of QoS control employed must be suitable to the variety of services offered over the communications network. This includes the ability to control QoS for diverse traffic requirements and support the introduction of new services with ease. The approach must be able to deal with a range of traffic characteristics such as traffic aggregation and traffic smoothing.

- **Maximise Resource Utilisation:** The network operator wishes to ensure that investment in infrastructure is warranted through ensuring available resources are being used to the maximum of their potential, within the constraints of SLA compliant service delivery.

- **Maximise Profit:** The network operator wishes to maximise resource utilisation while maintaining customer satisfaction, at minimal cost of operational and capital expenditure. The network operator strives to increase the total profit of the business through ensuring SLA compliant service delivery and optimal resource utilisation with minimal operational and capital expenditure.

In this thesis we show how our purely empirical effective bandwidth estimation approach can be used as input to performance optimisation strategies for QoS control of traffic within the network. We focus on aspects of two processes, namely: long-term network planning and short term admission control for IPTV flows.

1.1 Main Contributions

The thesis contributes to the area of QoS control within communication networks. It specifically addresses the estimation of effective bandwidth of real traffic flows with set QoS targets on packet delay, and uses such estimates in two performance optimisation strategies for QoS control: network planning and admission control. The main contributions are:

- A comprehensive examination and analysis of the performance of traffic model reliant effective bandwidth estimation approaches and justification for the use of a purely empirical effective bandwidth estimation approach (Davy *et al.*, 2006), incorporating:

  - The specification of a purely empirical effective bandwidth estimation algorithm and demonstration of its suitability for use within a communications network.
  - A comprehensive analysis of traffic model reliant effective bandwidth estimation approaches versus a purely empirical approach for use with real traffic flows under a number of scenarios involving varying traffic types, QoS targets and traffic aggregation.

- A process for supplying appropriate QoS related input to the network planning process at minimal operational and capital expenditure to the network operator (Davy *et al.*, 2007b,c), incorporating:

  - Specification of a process for the estimation of a demand matrix from available network accounting data, which is enhanced using effective bandwidth coefficients to be cognisant of QoS related characteristics of traffic.
  - An evaluation of the accuracy of such an approach and identification of its limitations with respect to providing appropriate QoS related information to the network planning process.
  - A comprehensive economic analysis of the proposed process against a direct estimation approach to estimating the demand matrix, justifying the former's effectiveness at minimising operational and capital expenditure for the network operator in supplying appropriate input to the network planning process.

- The proposition of two IPTV focussed admission control algorithms based on purely empirical effective bandwidth estimations (Davy *et al.*, 2007a, 2008), incorporating:

  – An evaluation of a number of traffic model reliant effective bandwidth based admission control algorithms with respect to their effectiveness in maintaining QoS targets of admitted traffic and bandwidth utilisation during periods of high network load.
  – The proposition and evaluation of two admission control algorithms based on purely empirical effective bandwidth estimation. The first algorithm employs a simple evaluation of whether there is sufficient bandwidth available to support the admission of a requesting service without impacting QoS performance. The second algorithm utilises information relating to the cost, duration and request frequency of specific IPTV content to prioritise flows that maximise revenue for the network operator at the time of admission.

1.2 Main Conclusions

The main conclusion of this thesis is that current performance optimisation strategies for QoS control can be improved through the use of a purely empirical method of effective bandwidth estimation. As a result, traffic engineering based performance optimisation strategies can more accurately provision bandwidth resources whilst ensuring QoS targets of traffic are maintained. The thesis demonstrates flaws with traffic model reliant approaches for estimating effective bandwidth and the negative effect using these approaches can have on the performance of the network. In detail the main conclusions are:

- Traffic model reliant effective bandwidth estimation algorithms are unable to capture the affect statistical multiplexing has on effective bandwidth of aggregated traffic flows. This has a direct impact on the dependability of performance optimisation strategies for QoS control of traffic. The reason for this is their reliance on traffic models that do not adequately model the effect of statistical multiplexing on the effective bandwidth of aggregated traffic flows.

- A purely empirical estimation of effective bandwidth is independent of any traffic model assumptions. The approach can accurately estimate effective bandwidth requirements of real traffic flows, and can also capture the affect statistical multiplexing has on the effective bandwidth of aggregated traffic flows. Such an approach ensures that performance optimisation strategies for QoS control employing these estimations efficiently use available bandwidth whilst maintaining QoS targets of traffic.

- Current methods for reducing the cost of supplying demand matrix information to the network planning process reduce the accuracy of the estimated demand matrix. These approaches can only collect a limited amount of packet level metrics such as packet delay, loss and jitter, as summarisation and sampling methods are employed. We outline a process of enhancing such a prepared demand matrix to ensure the information supplied to the QoS-aware network planning process is suitably accurate.

- Analysis of the process used to enhance the demand matrix through capturing a relationship between mean throughput from edge to edge of the network and associated effective bandwidth requirements, shows that the approach can estimate effective bandwidth requirements of traffic aggregates to within 10 - 15% accuracy, for the representative scenarios explored. We believe that this level of accuracy is viable for network planning as planning for future traffic demands is dependent on human behaviour patterns, which can not be predicted with a high degree of certainty.

- Based on an economic analysis, we find that a reduction in operational and capital expenditure of up to 80% can be gained by the network operator within the process of supplying QoS related demand matrix information to the network planning process through the use of our approach in comparison to a direct measurement approach.

- Using traffic model reliant approaches to estimate effective bandwidth for admission control strategies can lead to sub-optimal utilisation of bandwidth and control of QoS for admitted traffic during periods of high network load. By using a purely empirical approach for effective bandwidth estimation, our IPTV focused admission control algorithms, can guarantee that admitted traffic will maintain QoS targets at the point of admission to the network.

- We demonstrate that a revenue optimisation strategy can be employed within the IPTV focused admission control decision process, which can both ensure QoS targets of admitted traffic are maintained and maximise revenue over a period of high network load by prioritising flows that maximise revenue for the network operator.

1.3 Thesis Outline

Chapter 2 initially introduces concepts of traffic engineering within communications networks. The chapter introduces important traffic metrics and measurement architectures standardised by the Internet Engineering Task Force (IETF). It goes on to discuss a number of key areas in network management including the QoS control and network accounting. The particular scenario of IPTV is introduced and architectural and QoS considerations are discussed. This is followed by a discussion of specific aspects of performance optimisation strategies employed for QoS control in line with network operator requirements over different time scales. These areas are effective bandwidth estimation, QoS related demand matrix estimation in support of long term network planning and finally, admission control and its effectiveness at managing QoS control of traffic over the short term. The chapter concludes with a motivation for the work within this thesis through identification of open research areas to be addressed.

Chapter 3 specifies a purely empirical effective bandwidth estimation algorithm, and performs a comparison against two traffic model reliant effective bandwidth estimation approaches. The chapter identifies weaknesses with the traffic model reliant methods for usage within a communications network. It goes on to perform an evaluation of the purely effective bandwidth estimation algorithm, with regards to control variables that can impact the performance of the algorithm.

Chapter 4 addresses the issue of supplying appropriate input to the network planning process. In the chapter the QSPlan process that uses available accounting data to supply this information with the intention of reducing operational and capital expenditure for the network operator is specified. In QoSPlan a method of enhancing the demand matrix with QoS related information through the use of the effective bandwidth coefficient is utilised. The chapter finishes with a comparative economic analysis of both a direct measurement approach and the QoSPlan approach of establishing viable input to the network planning process.

Chapter 5 provides an in-depth analysis of the performance of hybrid admission control algorithms within the scenario of IPTV service delivery. The chapter identifies the failures of the reviewed algorithms with respect to QoS control over aggregated traffic flows and efficient bandwidth estimation during times of high network load. The chapter proposes two admission control algorithms based on empirical estimation of effective bandwidth that are designed to address the failures of the reviewed admission control algorithms. The rst algorithm employs a simple evaluation of whether there is sufcient bandwidth available to ensure, with an appropriate degree of condence, that QoS targets will not be violated if a requested ow is admitted. The second admission control algorithm proposes a method of revenue maximisation during times of high network load. The algorithm uses IPTV specific content information such as cost, duration and frequency of requests, to prioritise flows that maximise revenue to the network operator at the time of admission.

Chapter 6 summarises the main results and conclusions of the work, discussing the suitability of the purely empirical effective bandwidth estimation algorithm for deployment in a network performance optimisation strategy for QoS control within a communications network. The chapter finishes with a discussion of future work activities.

Chapter 2

Background Information and Literature Review

This chapter provides an overview of the state of the art regarding Quality of Service (QoS) control of communications networks. §2.1 discusses background information relating to traffic engineering in general, discussing two key areas central to traffic engineering: measurement and optimisation. The section finally introduces IPTV, discussing architectural and QoS considerations. With respect to these processes, §2.2 provides an in depth review of available literature regarding the areas we are particularly interested in; effective bandwidth estimation, demand matrix estimation for network planning and admission control. Finally §2.3 concludes this chapter with a summary of the work reviewed and a motivation for the work described in the remainder of this thesis.

2.1 Background Information

QoS control is defined as the process of ensuring packet level guarantees on the performance of traffic across the network (Campbell *et al.*, 1994). To ensure traffic is delivered to the customer within QoS constrained Service Level Agreements over a range of time scales, the network operator must perform various traffic engineering activities (Awduche *et al.*, 2002). This section gives an introduction to the basic concepts of QoS control within the communications network.

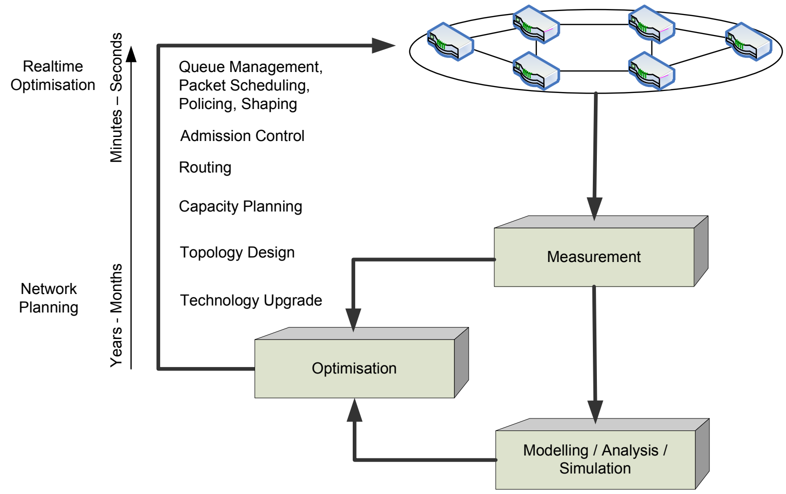

Figure 2.1: IETF Traffic Engineering (Awduche *et al.*, 2002).

2.1.1 Traffic Engineering

The IETF define traffic engineering as the aspect of network engineering that deals with the issue of performance evaluation and performance optimisation of the communications network (Awduche *et al.*, 2002). A core objective of the network operator is to achieve this as cost efficiently as possible (Gupta, 1985).

As can be seen in Fig.2.1, the process consists of three phases, measurement, modelling and optimisation:

- The process of measurement, as described in section 2.1.2.2, determines the current demand on the network over various time-scales. Measurement also serves as feedback into the optimisation process as indication on whether particular optimisation strategies are improving network performance;

- The modelling process is aimed at understanding the problem domain being optimised. Within the domain of traffic engineering, models are developed to gain an understanding of such elements as user behaviour, diurnal trends, as well as the actual traffic models within the network;

- Finally optimisation, can be considered as a range of steps, depending on the optimisation goal and the time frame. These steps may include long-term network planning activities such as ensuring adequate capacity is available for future customer demand, routing and topology design or short-term objectives such as queue management, packet scheduling and admission control.

We now discuss the aspects of these three phases relevant to this thesis: traffic modelling, long term network planning, short term admission control and effective bandwidth estimation.

2.1.1.1 Traffic modelling

Traffic modelling is a method of theoretically representing the behaviour of packet or flow level characteristics for a traffic source. A traffic model is quite useful for evaluation and proposition of various performance optimisation strategies within the network. Traditionally, packet arrival processes were modelled following a Poisson process, or using Markov Chains for more complex behavioural patterns. At a flow level, packet trains have been proposed as a method of modelling various traffic characteristics such as packet train size and packet train length (Jain & Routhier, 1986). This method has been successfully used to model traffic for the purpose of traffic classification, as discussed by Divakaran *et al.* (2006).

Research has shown that network traffic can exhibit rate variations over multiple time scales, leading to behavioural patterns such as self-similarity[1] within traffic. In such cases, traffic is quite difficult to model accurately. A wide range of publications have analysed this area highlighting the difficulties in modelling such traffic with general probability distributions; examples include: (Gong *et al.*, 2005; Leland *et al.*, 1994; Mansfield *et al.*, 2001).

A common method of detecting self-similarity is by measuring the Hurst[2] parameter of traffic. The Hurst parameter is non-deterministic in that it expresses what is actually observed within the data; it is not calculated so much as it is estimated. Therefore the measurement of self-similarity of traffic is dependent on the collection of data (or packet traces[3]) from the network. Within this case, we see how traffic measurements assist

[1]A self-similar object is exactly or approximately similar to a part of itself (Mansfield *et al.*, 2001). Within the domain of traffic rate variation, it relates to similar rate throughput at different time scales (seconds, hours, days).

[2]The Hurst parameter is referred to as the index of dispendence, and is the relative tendency of a time series to either strongly regress to the mean or cluster in a direction

[3]From our perspective, a packet trace is a time series of packet sizes, consisting of packet time inter arrival time in seconds and packet size in bits.

the development of a traffic model.

2.1.1.2 Optimisation

Network performance optimisation can be used to both remedy a problem that has occurred or that is incipient, or to improve network performance even when problems do not exist and are not anticipated. It is considered a continual process that usually consists of realtime optimisation and non-realtime network planning, the difference being the relative time scale and granularity of actions (Awduche *et al.*, 2002).

An objective of realtime performance optimisation is to control the mapping and distribution of traffic over the network with the objective of controlling congestion, assuring QoS guaranteed service delivery and optimising resource utilisation. Realtime optimisation deals with random incidents such as link failure or shifts in traffic demand, which can affect network performance over small to medium time scales (microseconds to minutes or hours), irrespective of how well a network is designed. Techniques for realtime optimisation include queue management, packet scheduling, and admission control. Network planning focuses on actions to evolve the architecture, technology, topology and capacity of the network systematically. This process is required to optimise the long term performance of the network, to support traffic growth and changes in traffic demands over time. Network planning and realtime performance optimisation processes operate together, to optimise network performance over both the short and long term.

A well-planned and designed network makes real-time optimisation easier, while a systematic approach to real-time network performance optimisation allows network planning to focus on long term issues rather than tactical considerations. Systematic real-time network performance optimisation also provides valuable inputs and insights into network planning. Core to both network planning and realtime network performance optimisation is the measurement of traffic demands over appropriate time scales. We now discuss traffic measurement in further detail and discuss the importance of this process with regards traffic engineering over different time scales.

2.1.2 Traffic Measurement

The measurement and collection of traffic metrics are central to any traffic performance optimisation process, whether this be long term network planning or short term admission control. This section discusses the topic of traffic metrics and measurement architectures in more detail.

2.1.2.1 Traffic Metrics

A primary method of evaluating the performance of a network is through the collection and analysis of various traffic metrics. The IETF IP Performance Metrics (IPPM) (Paxson *et al.*, 1998) working group has defined a set of such metrics for collection and analysis. The four primary performance metrics are bandwidth, delay, jitter and loss.

Bandwidth Bandwidth is defined as the volume of traffic that traverses a point (link or node), in a given period of time. Bandwidth is usually measured in bits per second (bps). A particular metric associated with bandwidth is Bulk Transfer Capacity (BTC), that defines the maximum transfer capacity between two nodes within a network (Mathis & Allman, 2001). BTC is calculated as:

$$BTC = \frac{data_{sent}}{time_{elapsed}} \tag{2.1}$$

One-Way delay metric The one-way delay metric is a measure of end-to-end packet delay, defined by the IETF IPPM (Almes *et al.*, 1999a) and is measured as the summation of a number of delay factors that contribute to delay of a packet as it traverses the network, specifically:

- **Transmission delay:** Time required to push all of a packet's bits onto the link.
- **Propagation delay:** Delay experienced by a packet traversing a digital circuit.
- **Queue delay:** Delay experienced by a packet waiting in a queue to be served.

Each of these factors can be minimised to reduce delay across the network. As transmission delay is dependent on link capacity, when a link is upgraded, transmission delay will be reduced for packets traversing that link. The same goes for propagation delay, which can only be reduced with a packet forwarding device hardware upgrade. Similarly, queue buffers can be adjusted dynamically to reduce queue-waiting times. The IETF IPPM also define a round trip packet delay metric (Almes *et al.*, 1999c) that includes a return delay time back to the originating node. Packet delay can degrade the performance of applications operating over the network and can also limit the total amount of bandwidth available within the network as less traffic is transferred over a period of time.

Jitter Packet jitter, or packet delay variation (Demichelis & Chimento, 2002), is the measure of the variation in packet delay between a particular set of packets from some base value such as mean packet delay. Packets taking alternative routes over the network, where the combination of delay factors can result in different one-way packet delays over different routes, generally cause packet delay variation.

Loss Packet loss is defined as the proportion of packets sent but not received at a node and is measured as a percentage or proportion of packets lost over total packets sent. The IETF IPPM has defined the one-way packet loss metric in (Almes *et al.*, 1999b) and also the packet loss pattern metric (Koodli & Ravikanth, 2002) to identify packet loss patterns within the network. Packet loss can be caused by queue buffer overflow, where packets are dropped at an interface when the queue buffer is completely full at the time of packet arrival. The occurrence of packet loss is a major concern within a heavily congested network as the probability of buffer overflow increases.

2.1.2.2 Measurement Architectures

The IETF has defined a number of network measurement architectures to support performance analysis and various other duties such as network accounting and auditing. These architectures are predominately based on passive measurement, where traffic traversing the network is collected for the purpose of monitoring and recording metrics. Active based network monitoring approaches inject packets into the network for monitoring purposes. This section focuses on the passive architectures developed by the IETF.

SNMP and RMON The Simple Network Management Protocol (SNMP) (Case *et al.*, 1990) was proposed by the IETF as a method of controlling network elements in a standardised fashion. The Remote Network Monitoring Architecture (RMON) (Waldbusser, 1991) was subsequently developed as an extension to this protocol to enable applications to gain access to network monitoring information, available within

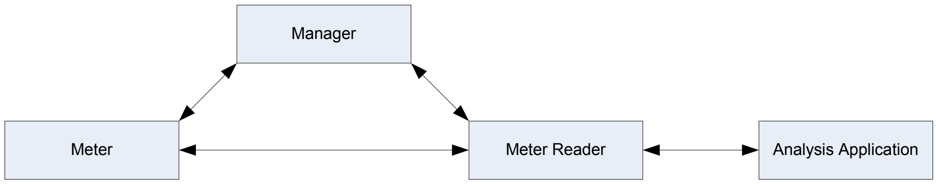

Figure 2.2: Real-time Traffic Flow Measurement (RTFM) Architecture.

the network elements. The objective of RMON is to enable efficient monitoring of network elements using dedicated devices such as network analysers, monitors and probes. RMON has a defined Management Information Base Module (Waldbusser, 1995) for SNMP.

Real-time Traffic Flow Measurement Architecture The Real-time Traffic Flow Measurement Architecture (RTFM) (Brownlee, 1999; Brownlee *et al.*, 1999) was developed to provide network operators with a standard method of monitoring traffic flows within the network. The architecture distinguishes between three measurement entities, namely a traffic meter, a meter reader and a manager, as illustrated in Fig. 2.2.

- **Traffic Meter:** Meter devices are placed at measurement points of interest within the network. Each meter selectively records network activity as directed by its configuration settings. It can also aggregate, transform and further process the recorded activity before the data is stored. The processed and stored results are called usage data.
- **Meter Reader:** A meter reader reliably gets usage data from meters and makes it available to analysis applications.
- **Manager:** The manager is an application that configures meter entities and controls meter reader entities. It uses the data requirements of analysis applications to determine the appropriate configuration for each meter and the proper operation of each meter reader.

The NeTraMet system is considered a successful implementation of the RTFM architecture (Brownlee, 1997, 2001).

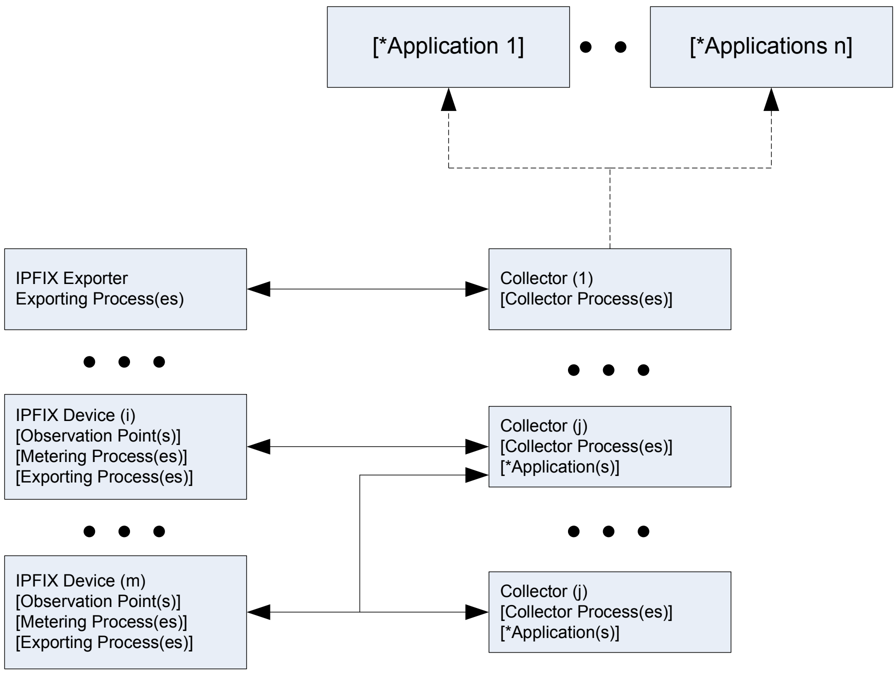

Figure 2.3: IP Flow Information Export (IPFIX) Architecture. (Claise, 2004)

IP Flow Information Export The IETF IP Flow Information Export (IPFIX) working group specifically addresses the issue of exporting traffic flow related information to applications in a standard way. The architecture defined in Sadasivan *et al.* (2006) proposes a standard method of collecting and exporting information related to IP traffic flows. Within this architecture a flow is defined as a set of packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. The properties of a flow can be defined as any combination or subset of the packet header fields (e.g. source address, destination address, or port number), or fields derived from packet treatment (e.g. next hop, or output interface). By defining properties of a flow, the architecture can collect a flow record representing anything from all packets passing an interface, to a single packet between two applications. The architecture defines a number of components, as illustrated in Fig. 2.3, described as follows:

- **Observation Point:** An observation point is a location within the network where packets can be observed.

- **Metering Process**: The metering process creates flow records from selected observed packets. The process uses selection criteria based on the properties of the flow records it is configured to collect. Packets can be selected for flow record inclusion based on two functions, the sampling function and the filtering function. The sampling function uses various sampling methodologies to choose a packet. The filtering function uses specified flow record properties to filter selected packets.

- **Exporting Process**: The exporting process sends flow records to one or more collectors using the IPFIX protocol.

- **Collector**: The collector decodes and stores flow records received from exporters, it also receives and stores control information used to interpret the collected flow records.

- **IPFIX Exporter**: A device that hosts one or more exporter processes is known as an IPFIX exporter.

- **IPFIX Device**: The IPFIX device hosts at least one exporter, one or more metering processes and one or more observation points.

Cisco have also developed a proprietary flow based network monitoring system, used by a wide range of applications, known as NetFlow 9 (Claise, 2004). NetFlow 9 and the current IPFIX specification have merged to define a single architecture definition within the IETF IPFIX Working Group (Claise, 2004). As a result, there are currently a wide range of applications built on top of NetFlow such as; NetFlow Analyzer (ManageEngine, 2008), NetUsage (Netusage, 2008), PeakFlow (ArborNetworks, 2008), FlowScan (Caida, 2008), Caligare Flow Inspector (Caligare, 2008) and NetFlow Tracker (FlukeNetworks, 2008).

Packet Sampling Packet sampling is a core feature of network monitoring architectures; for example, NetFlow and IPFIX use packet sampling to limit the volume of packets being processed into flow records. The objective of packet sampling is to reduce the volume of traffic being monitored and processed by the network monitoring system.

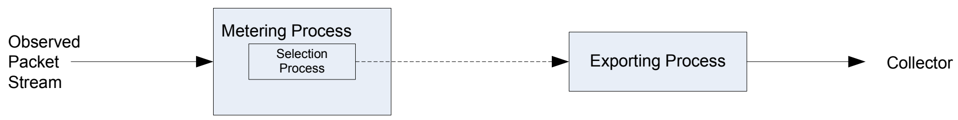

Figure 2.4: Packet Sampling (PSAMP) Architecture (Duffield, 2004).

The IETF Packet Sampling (PSAMP) working group has proposed a packet sampling and reporting framework (Duffield, 2004) aimed at standardising packet sampling procedures for network monitoring applications. The working group has also proposed a set of sampling and filtering techniques to be used within the framework (Zseby *et al.*, 2007).

The function of the PSAMP protocol is to select packets from a stream according to a set of standardised selectors, to form a stream of reports on the selected packets, and to export the reports to a collector. PSAMP selection operations include random selection, deterministic selection (filtering), and deterministic approximation to random selection (hash-based selection). Fig. 2.4 indicates the sequence of the three processes (Selection, Metering, and Exporting) within the PSAMP device.

Measurement probe placement The placement of measurement devices or probes within a network can impact on the types and values of metrics collected for analysis. There are three common locations within any single domain network topology that measurement probes can be placed: the core, the ingress and the egress. A single administrative network domain is commonly termed an Autonomous System (AS) .

- **Ingress:** The ingress of a network defines the entry point into an AS. Placing measurement probes at this location, facilitates collection of metrics of packets yet to be shaped and policed by the network (shaping and policing are discussed in section 2.1.3).

- **Core:** The core constitutes all nodes involved in packet forwarding within a single AS. Probes placed within the core can monitor forwarding behaviour within the network and collect metrics on how forwarding behaviour affects the performance of traffic metrics.

- **Egress:** The egress node defines the location where packets exit the AS to either enter another AS or be delivered to the end user. Metrics collected at this point can reveal details on packet shaping and the effect forwarding has on packets that have traversed the AS.

2.1.2.3 Traffic characteristics

Within a communications network, the characteristics of traffic can be divided into two categories: elastic and streaming. Elastic traffic relates to the transfer of digital objects through the network that can be transmitted at any rate up to the limit imposed by link and system capacity. The effect is that the bandwidth consumed by a particular flow will stretch and contract in an elastic fashion to suit availability. On the other hand, streaming traffic typically carries voice or video data and is characterized by a data transfer rate, which must be more or less preserved as the flow passes through the network. The effect here is that the traffic flow will only consume the required amount of bandwidth as needed and no more.

It is generally accepted that elastic traffic requires relaxed QoS targets on metrics such as packet delay and jitter, however packet loss is of a greater concern. For example, if we look at the operation of the Transmission Control Protocol (TCP), the loss of packet is seen as a sign of congestion by the protocol, and as a result the protocol will reduce the transmission rate. This has the effect of reducing response times within the associated application, as respective applications tend to wait for packets to be delivered before continuing.

Applications that consume real time streaming traffic only wait a limited period for a packet to arrive, before the packet is deemed irrelevant. This can be seen within Voice over Internet Protocol (VoIP) applications, where the processing of a late arriving packet could interfere with conversation quality. Table 2.1 demonstrates QoS requirements for various applications as stated in (Comer, 2008). The references to high, medium, and low are used to demonstrate metric priorities as apposed to strict QoS targets per application.

2.1.3 Quality of Service Control

The main objective of a QoS control architecture is to manage traffic metrics to ensure set QoS targets over traffic are maintained. Within the IETF, a number of standard

Table 2.1: QoS Performance Requirements of Applications.

Application	Bandwidth	Delay	Jitter	Loss	Traffic Type
VoIP	Low	High	High	Medium	Streaming
Video Conferencing	High	High	High	Medium	Streaming
Streaming VoD	High	Medium	Medium	Medium	Streaming
Streaming Audio	Low	Medium	Medium	Medium	Streaming
Client / Server Transaction	Low	Medium	Low	High	Elastic
E-Mail	Low	Low	Low	High	Elastic
File Transfer	Medium	Low	Low	High	Elastic

QoS control architectures have been proposed. We now discuss these architectures in detail.

2.1.3.1 RSVP

The Resource Reservation Protocol (RSVP), developed by the IETF (Wroclawski, 1997b), is used within various QoS control architectures such as IntServ, DiffServ and Multiprotocol Label Switching (MPLS) to transfer resource reservation requests. When an application is requesting the reservation of bandwidth within the network, the application will initiate a request. RSVP is the protocol used to carry this request to each node along the path within the network. RSVP attempts to reserve resources at each node to meet the requirements of the request. To make this request, the application must have explicit knowledge of the traffic requirements. This is specified in the form of a traffic specification (TSpec) that defines traffic requirements in the form of a token bucket specification. The token bucket system is used to access the conformance of a packet stream to a specified rule.

2.1.3.2 IntServ

The Integrated Service (IntServ) architecture (Braden *et al.*, 1994) was standardised by the IETF, to provide fine grained, flow based Quality of Service control of traffic within a network. IntServ specifies a set of functions on each node within the network that are designed to manage per flow traffic in a coordinated fashion. Nodes negotiate bandwidth reservations along a path using RSVP. The protocol details the bandwidth requirements of the requesting application, where each node along the proposed path must agree to satisfy this request if the connection is to be set up. The functional components that enable the IntServ architecture to control resources are as follows:

- **Reservation Setup Agent**: The reservation setup agent uses RSVP to negotiate and establish packet flow connections between other IntServ nodes.
- **Admission Control**: An IntServ node implements admission control on every RSVP request received. The admission control function implements decisions on whether a new flow can be admitted without impacting earlier commitments on QoS guarantees. Based on the TSpec submitted by the application, the admission control function on each IntServ node will admit or reject the request based on a number of criteria. The decision can be simply based on a summation of all accepted peak rate values of currently admitted traffic flows. If the addition of the new peak rate exceeds the maximum throughput capacity, the request is rejected.
- **Classifier**: Each incoming packet is mapped to a particular traffic class based on a set of packet related criteria. All packets in the same class will get treated the same by the packet scheduler.
- **Packet Scheduler**: The packet scheduler organises packets from different classes into appropriate queues. The queues are configured and prioritised to suit the type of traffic assigned to them. The packet scheduler will de-queue packets from the queues using various approaches such as Weighted Fair Queueing[1] or Priority Queueing[2].

Fig.2.5 defines the functional block of the IntServ architecture. Based on this architecture, IntServ defines two types of services that can be used to control QoS of traffic flows. These are the controlled-load service (Wroclawski, 1997a) and the guaranteed QoS control service(Shenker *et al.*, 1997b). Each uses a common method of requesting services within the network, however the traffic is treated differently. The controlled load service intends to guarantee best effort service (as experienced in a low

[1]Weighted Fair Queueing is a packet scheduling technique that uses a weighted algorithm to de-queue packets from multiple queues fairly.

[2]Within a multiple queue system, packets arriving at a priority queue will be de-queued first, thus ensuring traffic has the least likelihood of being rejected during times of high network load

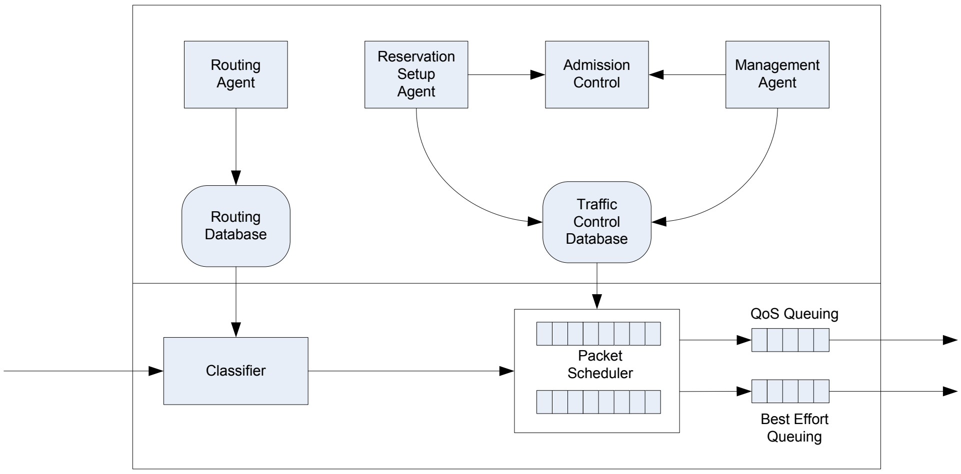

Figure 2.5: Integrated Services (IntServ) Architecture (Braden *et al.*, 1994).

loaded network) throughout the duration of the traffic flow. The service guarantees that conforming traffic will not experience delay greater than the maximum packet delay specified. Non-conforming traffic will be re-classed as best effort and forwarded through the network without QoS guarantees. The guaranteed QoS service provides firm bounds on end-to-end packet queueing delay. This approach employs a method of policing traffic at the edge of the network, to ensure all traffic within the flow is in conformance with the RSVP request to reserve the resources. Policing traffic at the edge consists of comparing traffic against the RSVP request and ensuring all traffic entering the network is in conformance. This ensures that no buffer overflows will be incurred, resulting in minimal loss of packets. The end-to-end behaviour provided by a series of network elements that conform to this service is an assured level of bandwidth that, when used by a policed flow, produces a delay-bounded service with no queueing loss for all conforming packets.

As IntServ is a flow based QoS architecture, each node is required to store information on each flow within the network. In the absence of flow aggregation, the amount of flow states stored on each node scales proportionally to the number of concurrent reservations, which can be potentially large on high-speed links.

2.1.3.3 DiffServ

The Differentiated Services (DiffServ) architecture (Blake *et al.*, 1998) is a course-grained class based QoS provisioning architecture developed by the IETF, stemming from a large body of research. DiffServ was developed for deployment over large-scale networks with the potential of hosting a high volume of concurrent traffic flows, directly addressing the shortcomings of IntServ. The objective is to aggregate traffic flows with common QoS requirements into predetermined traffic classes for transport through the network. DiffServ operates by policing traffic entering at the ingress of the DiffServ domain[1], aggregating traffic into a set of predetermined traffic classes through the use of packet classification, and specific common forwarding behaviours at each core node to treat packets within common classes identically.

Each traffic class has its own identification within the DiffServ domain, known as DiffServ Code Points (DSCP) (Nichols *et al.*, 1998). The DSCP is an 8 bit field within the IPv4 and IPv6 packet header which can take on a series of values, depending on the required forwarding behaviour of the packet. Forwarding behaviours of DiffServ nodes are termed Per Hop Behaviours (PHB). Fig. 2.6 denotes the functional components required by the DiffServ ingress node for traffic classification and conditioning. The following provides definitions for each of the functional components:

- **Classifier**: The classifier selects packets based on the content of the packet headers according to defined rules.
- **Meter**: Measures temporal properties of the traffic such as bandwidth, delay, jitter and loss. The traffic metrics measured by this device can affect the decisions of the marker, shaper and dropper, depending on their decision rules.
- **Marker**: The marker will set the DSCP of the packet based on defined rules. This process assigns packets to a particular traffic class. All packets within this class will experience the same forwarding behaviour through the DiffServ domain.
- **Shaper / Dropper**: The shaper will delay packets based on specified rules. The dropper will discard packets based on specified rules. These operations are

[1] A DiffServ domain is a contiguous set of DiffServ nodes which operate with a common service provisioning policy and set of packet forwarding behaviours on each node

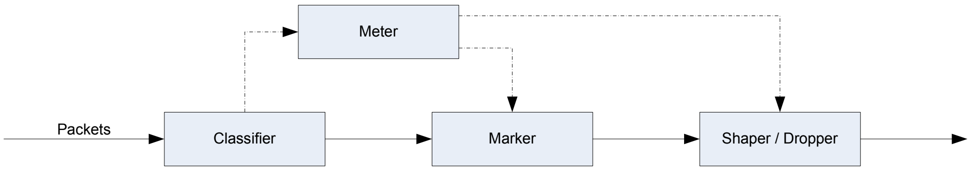

Figure 2.6: DiffServ Packet Classifier and Traffic Conditioner Function Blocks.

known as policing and can be applicable to different traffic classes based on the forwarding rules applied for that behaviour aggregate.

The PHB at each node, defines buffer management and packet scheduling policies on how packets within each traffic class are to be treated (Fig. 2.7). The objective is to ensure that PHB specifications are identical for each core node within a particular DiffServ domain. The network operator can then ensure that traffic fitting a specified profile will be classified into a traffic class and forwarded using the PHB across the network, ensuring QoS targets will be met. The traffic classes defined within DiffServ are: expedited forwarding (EF), assured forwarding (AF1 - 4) and best effort (BE). Each traffic class can be assigned a proportion of available resources at each node within the network along with its own PHB. Fig. 2.7 depicts a buffer manager using queues to differentiate between each of the traffic classes. The forwarding of packets are based on priority and weighted fair queueing to ensure that resources are partitioned reflecting the PHB specification.

Default PHB is essentially best effort service, taking the lowest priority in forwarding and packet scheduling. Expedited forwarding (EF) (Davie *et al.*, 2002) characterises a PHB ensuring low delay, low loss and low jitter. This behaviour group is suitable for real-time services such as voice and video. EF traffic is often controlled using priority queueing. EF is strictly controlled through admission control, and policing mechanisms to try to ensure no traffic overload is caused within the traffic class, causing a build up of queue delay inducing jitter and loss. Assured forwarding (AF) (Heinanen *et al.*, 1999) offers a means of providing different levels of forwarding assurance within a DiffServ Domain. Four classes are defined with the AF with three drop precedences within them, as can be seen in Table 2.2.

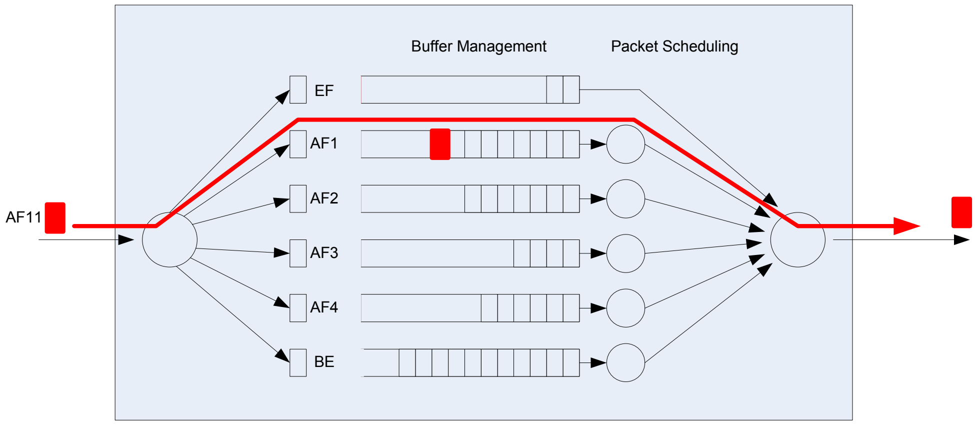

Figure 2.7: DiffServ Node Per Hop Behaviour (PHB) (Nichols *et al.*, 1998) .

Table 2.2: Assured Forwarding (AF) Behaviour Group.

	Class 1	Class 2	Class 3	Class 4
Low Drop	AF11	AF21	AF31	AF41
Medium Drop	AF12	AF22	AF32	AF42
High Drop	AF13	AF23	AF33	AF43

2.1.3.4 Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) was developed by the IETF, from work originally started by Cisco, before being submitted for standardisation (Rosen *et al.*, 2001). The theory is similar to that of ATM[1] and Frame Relay[2] operation, in that it provides a connection-oriented service for transportation of data across communications networks, using packet labelling to control how packets are routed through the network. This is achieved by setting the Forwarding Equivalence Class[3] (FEC) of the packet at the edge

[1]Asynchronous Transfer Mode is a cell relay, packet switching network and data link layer protocol which encodes data traffic in small (53 byte; 48 byte of data and 5 bytes of header information) fixed sized cells.

[2]Frame relay consists of an efficient data transmission technique used to send digital information quickly and cheaply in a relay of frames to one or many destinations from one or many end-points.

[3]Each router in a connectionless network makes an independent forwarding decision for a packet's next hop destination based on analysis of the packets header and some routing logic. Packets which result in the same forwarding decision are considered equivalent and are partitioned into a Forward Equivalence Class.

of the network, so no further header analysis is performed by the subsequent routers. MPLS defines two component routers used to perform packet labelling and switching, the Label Edge Router (LER) and Label Switching Router (LSR), along with the path that connects them known as the Label Switched Path (LSP)

- **Label Edge Router:** The LER is the router at the edge of the MPLS domain. The router is responsible for pushing labels onto the packet, defining the forwarding path through the network for that packet.
- **Label Switching Router:** The LSR is responsible for switching the labels used to route packets within an MPLS network. When the LSR receives a packet, it uses the label pushed onto the packet by the LER as an index to determine the next hop along the LSP and a corresponding label for the packet from a look-up table. The old label is then removed from the packet and switched with the new label before the packet is routed forward.
- **Label Switched Path:** The LSP is a dedicated unidirectional path through an MPLS network. The path is defined based on forwarding equivalence class definitions.

The IETF have also defined an approach for traffic engineering using MPLS and DiffServ architectures together in Faucheur & Lai (2003). The objective of this approach was to design LSPs for specific DiffServ traffic classes. The network operator can, for example, design an LSP following the shortest path through the network between two edge nodes, and reserve this path for EF traffic. Using this approach the network operator can engineer paths of guaranteed low latency for delay sensitive traffic and define alternate routes for more delay tolerant traffic. A basic depiction of this concept is demonstrated in Fig. 2.8.

2.1.4 Network Accounting

Network accounting was addressed by the Open Systems Interconnection Management Framework, who defined a process of collecting, interpreting and reporting, costing and charging oriented information on service usage (ISO, 1989). This process was divided into the following sub-processes: metering, pricing, charging and billing. Metering, as discussed previously, involves measuring and collecting resource usage information

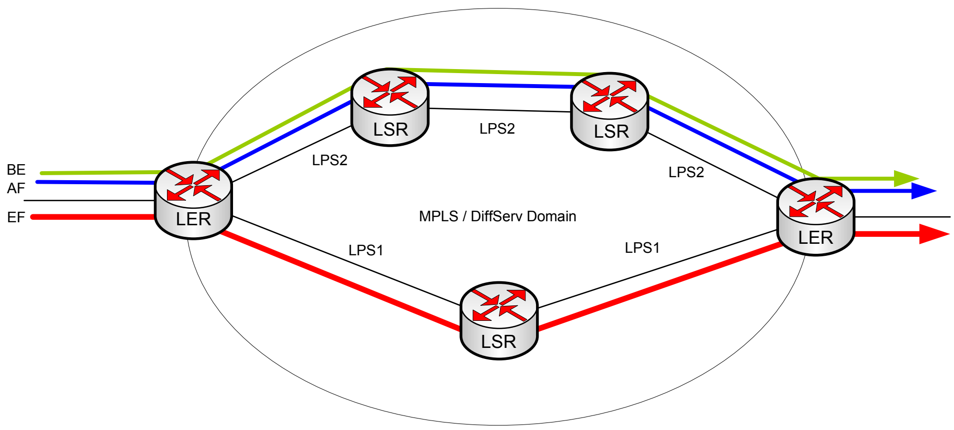

Figure 2.8: Traffic Engineering with MPLS and DiffServ. (Faucheur & Lai, 2003)

(such as IPFIX records). Pricing is the process of determining a cost per defined unit of resource usage. The charging process uses this cost per unit to translate the customer's resource usage information into an amount of money the customer has to pay. This amount is then used by the billing process to generate an invoice for the customer. Based on this definition, the IETF defined its own Internet based accounting architecture (Mills *et al.*, 1991), as depicted in Fig. 2.9.

The accounting server relies on the availability of metering information collected from network monitoring devices for the creation of session records. The session records detail the service interaction between the customer and the service provider over the entire session. Network devices transfer metering information to accounting servers using defined accounting protocols. Once records have been mediated with respect to pricing and charging, the records are sent on to billing servers, using some reliable transport protocol. The following sections discuss a number of the prominent accounting protocols for the transfer of accounting data between network devices and accounting servers.

2.1.4.1 TACACS+

The Terminal Access Controller Access-Control System Plus (TACACS+) protocol is an extension to the previously defined TACACS protocol used for providing access control to network servers and other networked computing devices (Carrell & Grant,

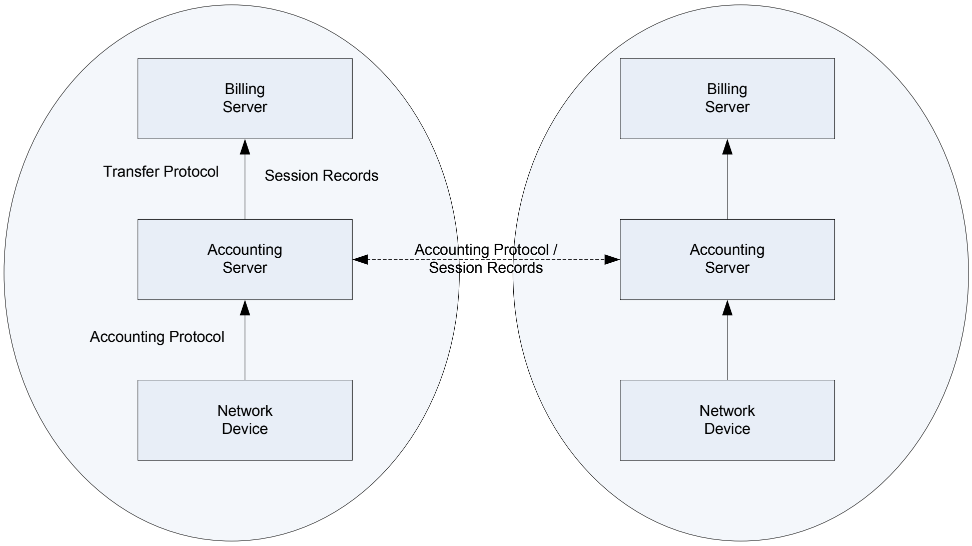

Figure 2.9: IETF Network Accounting Architecture (Mills *et al.*, 1991) .

1997). TACACS+ offers the additional feature of an accounting model with start, stop, and interim update messages. TACACS+ operates over TCP, supporting both transport and application layer acknowledgements, and is suitable for simultaneous usage control and handling of accounting events.

2.1.4.2 RADIUS

Remote Authentication Dial In User Service (RADIUS) Accounting (Rigney, 2000) was developed as an add-on to the previously defined RADIUS authentication protocol (Rigney *et al.*, 2000). RADIUS operators over UDP, so does not offer application layer acknowledgements or error messages, and therefore provides no protection against application layer malfunctions. It does, however offer a RADIUS Accounting-Response message, equivalent to a transport-layer acknowledgment. RADIUS does suffer from extensibility issues due to its limited command and attribute address space.

2.1.4.3 SNMP

The Simple Network Management Protocol (SNMP) has also been widely deployed in a variety of intra-domain accounting applications, typically polling network devices to collect usage data. Polling allows data to be collected on multiple accounting events simultaneously. Management applications are able to retry requests when a response is not received, providing resilience against packet loss or even short-lived network partitions. Implementations without non-volatile storage are not robust against device reboots or network failures, but when combined with non-volatile storage they can be made highly reliable. However SNMP based accounting systems generally require a large amount of processing and bandwidth resources, due to the polling activities and are generally seen as inefficient.

2.1.4.4 DIAMETER

The DIAMETER protocol was developed for the specific purpose of addressing the short comings of TACACS+ and RADIUS in providing Authentication, Authorisation and Accounting (AAA) for access technologies, such as wireless, Digital Subscriber Line (DSL), Mobile IP and Ethernet, routers and network access servers (NAS). The increased complexity and density was seen to put new demands on the AAA protocols and were deemed unsuitable for use within these domains (Calhoun *et al.*, 2003). The DIAMETER base protocol provides the minimum requirements needed for a AAA protocol, as required by Aboba, B. et al. (2000). It operates over either TCP or Stream Control Transmission Protocol (SCTP) (Stewart *et al.*, 2000). Due to its reliability, DIAMETER has been adopted into the 3rd Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS) architecture for AAA operations.

2.1.4.5 IP Detail Record (IPDR)

The IP Detail Record (IPDR) format (Cotton *et al.*, 2004) is an eXtentible Mark-up Language (XML) based accounting record format developed by the IPDR.org consortium. It was designed to provide information about IP based service usage and other activities to Operational Support Systems and Business Support Systems. The service provider, network/service element vendor and any other community of users determine the content of the IPDR record with authority for specifying the particulars of IP-based services. The specification of the IPDRDoc envelope in which the content is packaged, as well as encoding and transport protocols to exchange the information among systems is also produced by the IPDR organisation. IPDR record formats have been developed

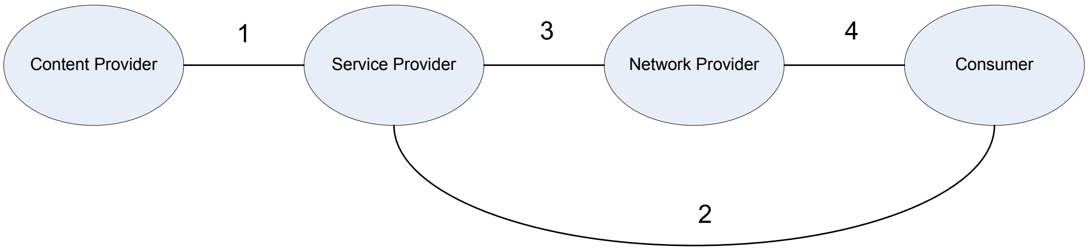

Figure 2.10: IPTV Logical Roles and Relationships (ATIS, 2007d).

for a wide range of IP based applications and services such as VoIP, Email, ASP, Public wireless access, wholesale bandwidth, digital content delivery and recently IPTV. IPDR records can be transported using either TCP or SCTP.

2.1.5 Internet Protocol Television (IPTV)

Internet Protocol Television (IPTV) is a term used to describe the delivery of services such as linear broadcast video content, unicast video-on-demand, and interactive services on video content (such as electronic program guides), over an IP network infrastructure. A standard IPTV architecture typically consists of a number of logical entities, namely the content provider, the service provider, the network operator and the consumer, as defined within the Alliance for Telecommunications Industry Solutions (ATIS) high-level architecture (ATIS, 2007d). As shown in Fig. 2.10, the content provider owns or is licensed to sell digital content over the IPTV network to service providers (1). The service provider is considered the main dealer of IPTV services to the consumer (2). A consumer is any subscribed customer of the IPTV service provider consuming services. The subscription the service provider offers the customer is bound by a Service Level Agreement (SLA). The SLA defines the quality of the service consumed by the customer, the service provider agrees to deliver. The service provider and customer are connected over the network operator's infrastructure (3), (4). The service provider and network operator also commit to an SLA.

2.1.5.1 IPTV Standardisation

There is a major push by various bodies to standardise aspects of the IPTV architecture. The following is a non-exhaustive list of such standards bodies that focus on various issues relating to IPTV.

- **ITU-T:** The International Telecommunications Union - Telecommunications Standardisation Sector (ITU-T) have set up an IPTV Focus Group to specifically look at the standardisation of an IPTV architecture. A number of documents form the base of their proposed architecture stemming from the Next Generation Network architecture developed in ITU-T (2004a) and ITU-T (2004b). A realisation of the proposed architecture has also been discussed in Lee & Knight (2005).

  The ITU-T have also standardised a video compression suite capable of meeting the high quality requirements of IPTV services in an efficient manner (ITU-T, 2005). It was written by the ITU-T Video Coding Experts Group (VCEG), together with the International Organisation of Standardisation (ISO) / International Electrotechnical Commission (IEC) Moving Picture Experts Group (MPEG), as the product of a partnership effort known as the Joint Video Team (JVT).

- **ETSI TISPAN:** The European Telecommunications Standards Institute (ETSI), Telecommunications and Internet Converged Services and Protocols for Advanced Networking (TISPAN) group are developing an architecture for the next generation network based on the 3GPP IMS specifications (Cedervall, 2008; Kisel, 2008; Mainwaring, 2007; Martti, 2007).

- **ATIS IPTV IIF :**The objective of the Alliance for Telecommunications Industry Solutions (ATIS) IPTV Interoperability Forum (IIF) is to enable the interoperability, interconnection and implementation of IPTV systems and services by developing ATIS standards and facilities related technical activities. The ATIS IIF group have currently published an IPTV architecture requirements document (ATIS, 2007b) and the IPTV Digital Rights Management (DRM) Interoperability requirements document (ATSI, 2006), along with continued work on IPTV QoS metrics and QoE models (ATIS, 2007a,c,d,e).

- **3GPP IMS:** The 3rd Generation Partnership Project (3GPP) has defined the IP Multimedia System (IMS) which can support advanced communications services such as those of IPTV related services. The 3GPP are working with other standards bodies such as ATIS and ETSI to develop IPTV service specifications for the IMS architecture.

- **TMOC:** The Telecom Management and Operations Committee (TMOC) are focusing on specifying an IPTV Operations Support System (OSS) based on the enhanced Telecom Operations Map (eTOM) defined by the TM-Forum (TM-Forum, 2005).

- **DSL-Forum:** The DSL-Forum has prepared a technical report on the issue of QoE requirements for triple play IPTV services (Rahrer *et al.*, 2006). The objective of the document is to present the recommended minimum end-to-end QoE requirements in terms of objective engineering measurements for IPTV services delivered to the customer.

2.1.5.2 IPTV Logical and Physical Topology

There is a general consensus between standards bodies including ATIS, ESTI, ITU-T, DSL-Forum and IPDR, that the logical topology of the IPTV network consists of a number of administrative domains: Super Head End, Video Hub Office and Video Serving Office, as shown in Fig. 2.11. Each administrative domain is responsible for management of various IPTV related service offerings to its respective customer base. These domains may belong to a single entity or multiple entities. The logical topology of administrative domains within an IPTV network are defined as follows:

- **Super Head End (SHE):** The SHE is the location for acquisition and aggregation of national level broadcast TV (or linear) programming. SHEs are also the central point of on-demand content acquisition, management, encoding, organisation and insertion.

- **Video Hub Office (VHO):** The VHO defines the video distribution points within a demographic market area (DMA). National content is received from each SHE. Local content is acquired and encoded at the VHO. VOD servers and other application servers are typically located in the VHOs. Insertion of local

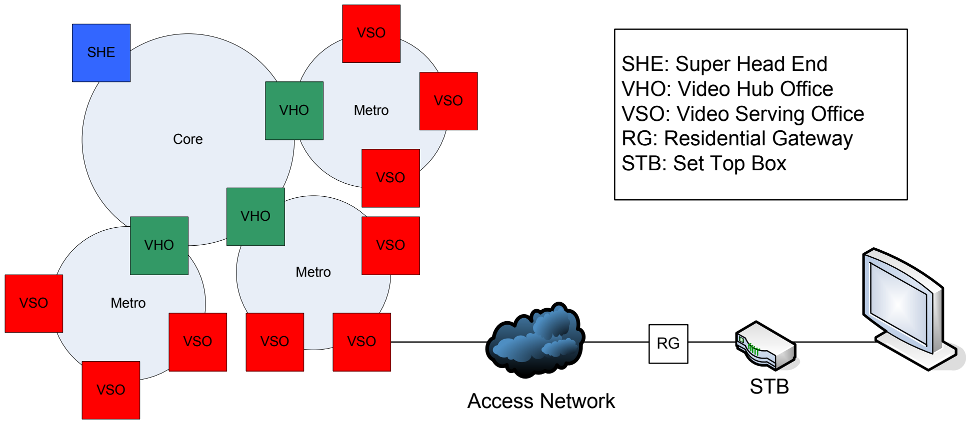

Figure 2.11: A high level overview of the IPTV logical distribution topology (Cotton & Kleinmann, 2006).

advertising is also performed in the VHO. IPTV services are provided from the VHO via the aggregation/ access network.

- **Video Serving Office (VSO):** The VSO contains/hosts all access systems used to connect all access systems to the subscribers. In addition, the VSO contains aggregation equipment to enable efficient and reliable interconnection to the VHO.

- **Residential Gateway (RG):** The RG node is dedicated to a single subscriber / household providing traffic management and routing between the access network and the home network. The RG function may be integrated with the network termination. The RG is a trusted device and is managed from the network.

Cisco have proposed and realised the following IPTV network architecture based on the Next Generation Network (NGN) (Lee & Knight, 2005). However the delivery of IPTV related services do not strictly require an NGN deployment. For example, Utopia have developed an IPTV based network for the delivery of triple play services to customers over a cable broadband network (Moerman *et al.*, 2005).

2.2 Literature Review

This section provides an analysis of published literature regarding those aspects of QoS control of traffic within a communications network related to the work within this thesis. §2.2.1 offers a review of approaches proposed to estimate the effective bandwidth of traffic flows. The objective of this review is to evaluate the suitability of such approaches for use within a real communications network. §2.2.2 discusses a number of approaches for estimating the demand matrix of a network. The approaches are reviewed based on deployment costs and attainable accuracy. Finally in §2.2.3 typical admission control algorithms used to control QoS of traffic with the network are reviewed.

2.2.1 Effective Bandwidth Measurement

The term *effective bandwidth* refers to the minimum amount of bandwidth required by a traffic flow to maintain a specified QoS target[1] (Kelly, 1996). Fig. 2.12 demonstrates the basic concept of effective bandwidth for given a traffic flow. As can be seen, the effective bandwidth lies between the peak and mean rate of a traffic flow (assuming the peak rate was found using an appropriate measurement interval). The reason for this is that if the peak rate was reserved for a traffic flow, no packets within the flow would experience queuing delay. However as QoS constraints over packet delay are imposed, the required amount of bandwidth necessary to transport traffic to within these QoS constraints reduces. The objective of using an effective bandwidth estimation algorithm is to accurately identify the required amount of bandwidth for a traffic flow to ensure QoS targets are maintained, without over provisioning resources.

A particular issue within a communications network is the effect statistical multiplexing of traffic at the point of aggregation has on the effective bandwidth of aggregated traffic flows specifically as traffic flows are aggregated, the variation in throughput is smoothed out within the aggregated traffic flow. For example, a period of low throughput from one traffic flow, may be counteracted by a burst in activity in another traffic flow. With respect to the measure of effective bandwidth, as traffic flows are aggregated, the effective bandwidth requirements of each traffic flow reduces as the level of

[1] A QoS target defines a limit on traffic metric values of a traffic stream. A QoS target on packet delay could be considered as: only 0.01% of packets within a traffic stream can experience packet delay of 100 ms or greater (Ferguson & Huston, 1998).

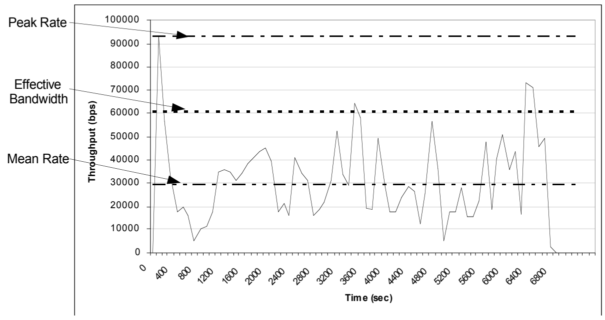

Figure 2.12: Demonstration of Effective Bandwidth.

aggregation increases. This has been observed by Botvich & Duffield (1995) and Simonian & Guibert (1995). Effective bandwidth estimation algorithms must be able to capture this effect if traffic performance optimisation strategies employing these algorithms are to effectively control QoS of traffic whilst minimising bandwidth utilisation. The following sections discuss a set of common approaches used to estimate effective bandwidth with respect to their suitability of implementation within an operational communications network.

2.2.1.1 Traffic Model Based estimation

Kelly (1996) proposed a theoretical framework for the prediction of effective bandwidth of a defined traffic source[1]. The formal definition of effective bandwidth for a traffic source is as follows:

$$\alpha(s,t) = \frac{1}{st} \log E[e^{sX[\tau,\tau+t]}] \qquad 0 < s,t < \infty. \tag{2.2}$$

where s is the space scale and t is the measurement interval. s may be measured in units of bytes^{-1} and t may be measured in seconds. $X[\tau, \tau+t]$ is the workload arriving in the period $[\tau, \tau+t]$ and the expected workload value is taken over the distribution of such random time periods. Estimation of the effective bandwidth of a traffic source also takes into consideration the resource over which the traffic is being transported. The

[1]The term 'traffic source' is used to describe a complete specification of a traffic model (Kelly, 1996). See appendix sectionA.3 for further explanation of how traffic models are generally defined.

appropriate s and t scales are therefore determined by the characteristics of the resource such as the capacity, buffer length and the required QoS targets. Kelly produced a number of studies based on this framework to measure the effective bandwidth of different traffic sources, including periodic sources including general on-off sources, fluid sources, Gaussian sources, and Lévy processes including compound Poisson sources and infinitely divisible sources [1].

Kelly notes that the theoretical approach he proposes cannot be deployed on an operational network without a complete description of all traffic sources. As this has been shown to be quite a challenging task (Gong *et al.*, 2005; Leland *et al.*, 1994; Mansfield *et al.*, 2001), alternative approaches have been developed to use static traffic model assumptions estimated from various traffic metrics as input to the effective bandwidth estimation algorithm. These algorithms include: the direct estimator (Gibbens, 1996); the block estimator (Duffield *et al.*, 1995); and the Kullback-Leibler Distance estimator (de Veciana *et al.*, 1995). The direct estimator, proposed in Gibbens (1996), applies the temporal average instead of considering the probabilistic expectation of the underlying traffic model process in equation 2.2. The block estimator, discussed in Duffield *et al.* (1995), in contrast to the direct estimator, considers non-overlapping blocks of arrivals over a time interval. The estimator is based on the assumption that the block arrivals are realisations of an independent and identically distributed random variables. This assumption limits the applicability of the estimator to the types of traffic that conform to such assumptions such as short range dependent traffic[2]. The Kullback-Leiber distance estimator, (de Veciana *et al.*, 1995), is based on a virtual buffer method, derived from equation 2.2. This approach also is only applicable to stationary, short-range dependent traffic sources.

2.2.1.2 Roberts's estimation

A flow-oriented approach proposed by Roberts *et al.* (1996) employs a theoretical prediction model. This algorithm estimates the maximum link utilisation threshold ρ_{max} required to maintain set QoS targets on packet delay. The approach models the traffic

[1] See appendix section A.2 for further explanation of distribution processes.

[2] Traffic with short range dependence suggests that coupling between rate values at different time scales decreases rapidly as the time difference increases. It has been shown that Internet traffic tends to demonstrate long-range dependence, see for example Gong *et al.* (2005).

arrivals on the link as an $N \cdot D/D/1 - \infty$ queue, with N homogeneous flows, each sending packets of expected packet size $E[B]$ and inter-arrival time $E[A]$. This approach uses simplistic traffic model assumptions at the core of its estimation process, therefore it is not suitable for use within domains where traffic does not conform to the proposed flow arrival model. This approach only measures effective bandwidth for a link as apposed to per traffic flow. This measurement defines the maximum proportion of bandwidth of the link that can be consumed by traffic requiring particular QoS targets to be maintained, before QoS violations occur. The algorithm takes into consideration the maximum number of flows that adhere to the defined traffic model, allowed on the link. As the traffic model used to estimate this threshold is static, any variations in traffic aggregation are not considered within the estimation.

2.2.1.3 Gaussian estimation

An approach proposed by Guérin *et al.* (1991) recognises the issue of effective bandwidth for aggregated traffic. To address this issue, Guérin proposes two approaches of estimating effective bandwidth; one is specifically designed to measure effective bandwidth requirements of a single traffic flow, while the other addresses estimation of effective bandwidth for aggregated traffic. The latter approach is based on the premise that as traffic is aggregated at a point, the distribution of the traffic arrival bit rate can be accurately modelled using a Gaussian distribution. Based on this assumption the author proposes to use standard approximations to estimate the tail of the bit rate distribution. It has been shown by Guérin *et al.* (1991) that as traffic flows are aggregated, the arrival bit rate does approach a Gaussian distribution, however this approach does not account for variation in traffic aggregation, impacting on the effective bandwidth estimation at various aggregation levels.

2.2.1.4 Empirical estimation

Empirical estimation of effective bandwidth attempts to overcome the limitations of traffic model based effective bandwidth estimation algorithms. The approach is based on the analysis of traffic being replayed through a simulated queue model. The approach observes the behaviour of the modelled queue buffer as the traffic is processed to measure an effective bandwidth value.

A method proposed by Liu & Baras (2004) involves collecting a packet trace from the network at a point where the effective bandwidth estimation is required. The approach simulates a FIFO queue with an adjustable queue service rate. The packet trace is processed through the FIFO queue at specified service rates, to measure the associated proportion of QoS target violations. The algorithm is based on the observation that as the service rate of the queue increases, the proportion of violating traffic decreases. The FIFO queue service rate for successive iterations is controlled by a search algorithm that decide when an appropriate queue service rate is found; this rate produces an appropriate level of QoS violations from the queue for the processed packet trace. This service rate is then taken as the effective bandwidth. Liu & Baras (2004) do not discuss the FIFO queue model used for this simulation, although FIFO queue algorithms are commonly simulated using packet level models. However a more efficient method of FIFO queue simulation is through the use of a fluid (continuous) model, Liu *et al.* (2001) demonstrated improvements in simulation execution time in comparison to packet level FIFO queue models.

The approach also estimates the optimal size of the packet trace to be analysed for effective bandwidth estimation. The justification within the paper is that too large a packet trace can result in an overestimation of effective bandwidth. The approach is independent of any traffic model assumptions and has been demonstrated to accurately estimate the effective bandwidth of aggregated traffic. As there is a certain amount of processing of the packet trace to find the effective bandwidth, the approach is implemented as an off-line estimation of effective bandwidth.

2.2.2 Demand Matrix Estimation

The demand matrix is a pair wise, edge-to-edge, matrix of traffic volumes that have traversed the network over a period of time (Medina *et al.*, 2002). The network demand matrix is considered a core element of the network planning process. Traditionally, the demand matrix is established by analysing traffic within the network directly, through the use of dedicated network monitoring devices (probes). Such approaches can establish a highly accurate traffic demand estimation across the network. However, as additional hardware for these probes require installation, operation and maintenance, this approach tends to incur high costs to the network operator, increasing both operational and capital expenditure. Core to the ethos of network planning is ensuring cost efficient and timely planning decisions are made in line with network operator objectives.

This section evaluates a number of proposed approaches developed as cost efficient methods of establishing the demand matrix for a single operator domain. Direct measurement approaches are discussed in §2.2.2.1. A packet sampling based approach used to infer the demand matrix while reducing data analysis volumes is discussed in §2.2.2.2. Finally, §2.2.2.3 evaluates a number of approaches used to infer the demand matrix from available accounting data. The objective of this section is to highlight the sacrifice in QoS control the network operator makes when introducing methods with the aim of reducing operational and capital expenditure in the supply of demand matrix information to a network planning process.

2.2.2.1 Direct monitoring

Direct monitoring systems provide the network operator with real time monitoring information of the network infrastructure in a non-intrusive manner. To achieve this, additional probing devices are usually deployed within the network at various locations. There are a wide range of commercial network monitoring systems such as: Network Instrument's Observer (Network-Instruments, 2008), Object Planet's Network Probe (ObjectPlanet, 2008), Gigamon System's GigaVUE-MP (GigamonSystems, 2008), ClearSight's Analyser (ClearSight, 2008). These systems can collect a huge amount of network activity information, such as complete packet trace copies of traffic traversing the network. For this reason, additional storage is also required within the network to host this measurement information. The demand matrix can be estimated from such systems with a high degree of accuracy; however, as such systems require the installation of additional hardware, they can be quite expensive to deploy and operate.

2.2.2.2 Trajectory sampling

Trajectory sampling (Duffield & Grossglauser, 2001) defines a direct measurement approach to monitoring network activity. However, this approach was designed to reduce the amount of traffic required to estimate a network wide demand matrix. The method was developed with the objective of reducing the operational cost for the network operator while maintaining a high level of accuracy. The method samples packets that traverse each link (or subset of links) within a measurement domain[1]. The authors propose a formal approach to filtering and selecting packets based on hash functions of the packets' contents. As the approach uses identical hash functions at all measurement points across the network for packet selection, the approach can monitor a subset of traffic traversing the network. With this information, the trajectory of traffic traversing the network can be inferred, as well as packet delay, packet loss and of course a preparation of a network wide demand matrix. This method can be relatively low cost to deploy within the network, and the measurement traffic is modest and can be controlled precisely.

2.2.2.3 Accounting based inference approaches

Feldmann *et al.* (2001) propose an approach to estimating a network wide demand matrix from IP Flow records collected at ingress points in the network. The general approach is as follows. For each flow collected at an ingress point, its destination address is mapped to an egress node at the edge of the network. The approach assumes that no traffic is consumed within the core of the network. Therefore all traffic entering an ingress point has a corresponding egress exit point. The approach also makes the assumption that the volume of demand within the flow is uniformly distributed from start to finish. Based on this assumption the volume of the flow is divided into equal bins of set durations. The volume within each bin is added into the demand matrix specifying the demand between the ingress and egress node for that bin period. Once all flows are processed, the demand matrix will contain total traffic demand between ingress, egress pairs over each bin period. The authors also state that additional information such as protocol and type of service information held within the flow record can be used to enhance the demand matrix information.

Papagiannaki *et al.* (2004) proposes an enhancement to this approach. The work presents a method of distributing the operation of demand matrix estimation among the ingress router nodes. The approach focuses on distributing two essential functions core to the estimation of the demand matrix from flow records and makes the following recommendations:

[1]Within this context, the measurement domain is the part of the network where trajectory sampling is deployed.

- Implement a function to map destination network prefixes to egress links or routers within the domain;
- Modify the definition of the flow record in order to include the result of this mapping.

Based on the analysis performed within the paper, the authors found that up to 99% communications overhead can be reduced if the proposed approach was deployed over a traditional direct measurement approach.

2.2.3 Admission Control Techniques

Admission control is a strategy usually employed to control access to a resource in order to maintain some stated objective. This objective can include: maintaining a secure system through firewall policies and access control lists; avoiding congestion on a network node through the use of the admission control algorithms such as that by Turán *et al.* (1998); load control over an entire network, as demonstrated in Jennings *et al.* (2001); and maintaining stated QoS targets of traffic over limited bandwidth, such as the work of Georgoulas *et al.* (2005b) and Milbrandt *et al.* (2007). This section focuses solely on the latter objective of maintaining QoS of admitted traffic flows during periods of high network load within an operational network scenario.

As stated in Shenker *et al.* (1997a), admission control within the IntServ and DiffServ network is performed based on the TSpec within the RSVP request. The TSpec contains an *a priori* description of the traffic requirements, stating a peak throughput rate. The admission control logic would base its decision on a comparison of total peak rates of all admitted services to the total capacity available of the link. If adequate bandwidth is available, the service can be admitted. This type of admission control algorithm is generally known as Parameter Based Admission Control (PBAC). This approach can lead to over provisioning for admitted service requests, as the peak throughput of a request is rarely consumed for a duration of the session.

A number of approaches have been developed to address the issue of over provisioning by basing decisions on live traffic measurements. Measurement based admission control (MBAC) attempts to learn the characteristics and requirements of the traffic flows admitted and bases future decisions on this knowledge. Several detailed evaluations of various measurement based admission control algorithms have been published, for example: Jamin *et al.* (1997); Turán *et al.* (1998) and Breslau *et al.* (2000). The advantage of using measurement-based approaches over parameter based, is that no *a priori* knowledge of the service request is required, therefore a greater utilisation of available resources is achievable.

2.2.3.1 Measurements and *a priori* traffic descriptor based admission control

The third approach we focus on is hybrid admission control. Hybrid approaches depend on both a description of the requesting service along with detailed traffic measurements to aid the decision process. The following section discusses two representative hybrid admission control approaches with respect to QoS control and bandwidth utilisation.

The Measurements and *a priori* traffic descriptor based admission control algorithm, proposed by Georgoulas *et al.* (2005b) focusses on ensuring QoS targets of admitted traffic are maintained while maximising bandwidth utilisation. This approach uses the estimation of effective bandwidth to control bandwidth utilisation of admitted traffic. The technique it uses for this estimation is based on the the theory of equivalent capacity (Guérin *et al.*, 1991). The approach uses both the traffic descriptor defining the peak rate of the requesting service and measurements of admitted traffic within the network to make admission decisions. Measurements of the mean throughput of the admitted traffic, along with the standard deviation of the aggregate bit rate, are used as input to estimate effective bandwidth requirements of the admitted traffic.

The main disadvantage of this approach stems from its reliance on traffic model based effective bandwidth estimation. As stated by Guérin *et al.* (1991), the approach depends on a strict traffic model assumption, which states that as traffic is aggregated the stationary bit rate approaches a Gaussian distribution. If traffic is not adhering to this strict model, estimations of effective bandwidth can be inaccurate, thus reducing the control over QoS the algorithm needs to maintain.

2.2.3.2 Experience Based Admission Control

Experience based admission control (EBAC) is another hybrid admission control algorithm, proposed by Menth *et al.* (2004); Milbrandt *et al.* (2006, 2007). The algorithm focuses on ensuring accurate and timely predictions of bandwidth requirements in the face of highly fluctuating traffic. The approach depends on a theoretical prediction of

Table 2.3: Summary of effective bandwidth estimation approaches.

Method	Traffic Model	Statistical Multiplex	Online
Traffic model based	Yes	No	No
Roberts estimation	Yes	No	Yes
Gaussian estimation	Yes	No	Yes
Empirical estimation	No	Yes	No

effective bandwidth and uses this estimation to set a maximum link utilisation threshold. This threshold defines the maximum allowable throughput on the link, before QoS violations in packet loss targets begin to occur. The algorithm bases this estimation on the approach developed by Roberts *et al.* (1996) (as presented in section 2.2.1.2). The authors conclude that the algorithm is highly adaptive and can guarantee QoS will be maintained even with high variability of admitted traffic and burstiness. This approach only measures the maximum allowable proportion of link bandwidth that can be used to ensure QoS targets are guaranteed on traffic fitting a statically defined traffic model. Therefore it can not accurately estimate effective bandwidth requirements for traffic not adhering to this traffic model definition. This has a direct impact on the actual control over QoS the algorithm has, during scenarios where traffic flow characteristics can change, in particular where there is statistical multiplexing of traffic flows.

2.3 Summary and Conclusions

This chapter began by providing background material related to operation and control of QoS within the communications network. This covered issues relating to the collection and measurement of traffic performance metrics. Performance analysis, control and accounting of these metrics were also covered. Following this, a discussion of tools and techniques commonly used for the control of QoS within the communications network was presented. This included effective bandwidth estimation, demand matrix estimation and admission control. We define each part of the work within this thesis as an aspect of the traffic engineering control loop Fig. 2.13.

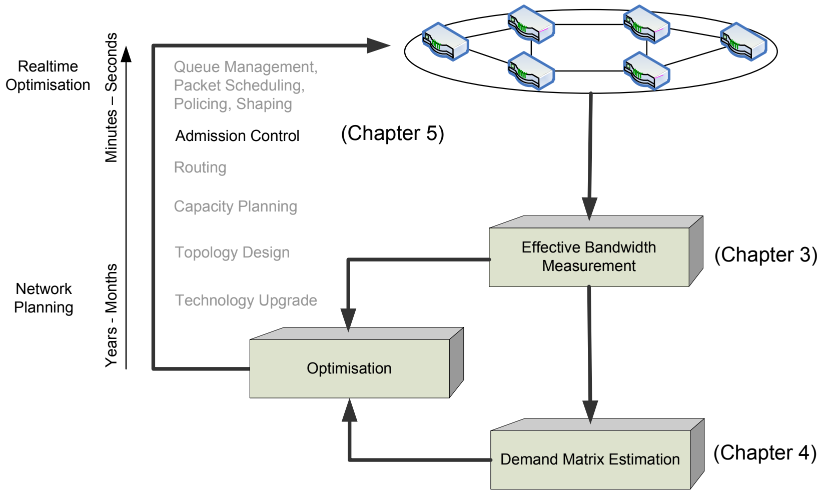

Figure 2.13: Measurement and Analysis Aspects.

2.3.1 Analysis of effective bandwidth estimation approaches

Based on the review it is clear that traffic model based estimation of effective bandwidth is highly dependent on either the accurate modelling of traffic, or static assumptions on traffic behaviour. It is our contention that such approaches are not suitable for accurate effective bandwidth estimation within an operational network scenario, and we therefore believe an empirical approach to effective bandwidth estimation is superior, particularly with respect to suitable for addressing operational network scenarios such as statistical multiplexing within traffic.

2.3.2 Analysis of demand matrix preparation approaches

Based on the analysis of the proposed demand matrix estimation approaches, we have compiled a summary of our conclusions in Table 2.4. This analysis was based on each approach's ability to supply accurate information of network demand and associated QoS metrics to a network planning process. It evaluates each approach based on its potential impact on the network operator's operational and capital expenditure. As can be seen in Table 2.4, as the operational and capital expenditure of a network monitoring system decreases so too does the level of accuracy and ability to capture vital QoS metrics such as packet level loss, delay and jitter. The ideal objective of the network operator is to supply accurate, QoS related demand matrix information to the network

Table 2.4: Summary of demand matrix estimation approaches.

Approach	OPEX	CAPEX	Demand Matrix accuracy	QoS Metrics
Direct Measurements	High	High	High	High
Trajectory Sampling	Medium	Medium	Medium	Medium
Accounting Based Inference	Low	Low	Low	None

planning system at minimal cost. To ensure this is achieved, lower cost deployments must be appropriately enhanced to capture QoS related characteristics of traffic. In this thesis we outline a method of enhancing the accounting based demand estimation process with appropriate empirically measured effective bandwidth estimations.

2.3.3 Analysis of admission control algorithms

The objective of admission control is to ensure customers' traffic flows are allocated sufficient bandwidth to ensure service level agreement constraints relating to packet level QoS targets are maintained during periods of high network load. A key feature of any admission control algorithm is how well it predicts the level of bandwidth required to admit a requesting service. If a new flow is accepted, the associated packet level QoS targets should be met, without affecting QoS of already admitted flows. Fundamental to achieving this is the accurate prediction of required effective bandwidth of aggregated traffic already admitted to the network.

We have reviewed a number of admission control algorithms, with the objective of maintaining packet level QoS targets of traffic during periods of high network load. The review discussed a number of admission control algorithms that attempt to guarantee packet level QoS through over provisioning (PBAC) or maximise bandwidth utilisation through traffic measurement and modelling (MTAC and EBAC). Table 2.5 outlines a summary of our review. As PBAC over provisions for admitted services, the gain of guaranteed QoS control can come at the cost of resource under utilisation. This can have a direct impact on the volume of customers admitted to the network, thus impacting the level of potential revenue. However, as both MTAC and EBAC rely on

Table 2.5: Summary of admission control approaches.

Approach	Bandwidth Utilisation	QoS Control
PBAC	Low	Guaranteed
MTAC	Medium	Unstable
EBAC	Medium	Unstable

traffic model assumptions for the estimation of effective bandwidth, their estimation of effective bandwidth can only be dependable under traffic conditions in line with their assumptions. We contend that such approaches are inappropriate for admission control within an operational network scenario, and can lead to poor utilisation of bandwidth within the network.

2.3.4 Concluding Remarks and Hypothesis

The objective of the network operator is to control the QoS of services through deployment of traffic engineering strategies in a cost efficient and timely manner. A fundamental tool used to ensure this objective is maintained is the accurate estimation of effective bandwidth requirements of traffic. Clearly an accurate estimation of effective bandwidth can ensure optimal usage of available resources within the network while maintaining packet level QoS targets on traffic. With respect to minimising operational and capital expenditure incurred, this section focused on analysing attempts to reduce the cost of demand matrix preparation. Ideally the network operator wishes to establish a QoS aware view of network demand, but with minimal investment in network infrastructure.

Based on this literature review, the following issues have been identified:

- Effective bandwidth estimation algorithms reliant on traffic model assumptions cannot account for a wide range of traffic types that do not adhere to the traffic model assumptions. This has a direct impact on the dependability of the effective bandwidth estimations and their usefulness in the deployment of traffic engineering strategies for QoS control of traffic.

- In attempting to reduce operational and capital expenditure of supplying vital demand matrix information to long term network planning processes, proposed approaches sacrifice accuracy in measuring traffic demands and packet level metrics such as packet delay, loss and jitter. For such approaches to be effective in QoS control by the network planning process, the demand matrix information requires QoS related enhancements.

- Admission control algorithms that attempt to improve bandwidth utilisation in comparison to peak rate over-provisioning, sacrifice QoS control as proposed approaches use effective bandwidth estimation algorithms based on traffic model assumptions.

- Admission control algorithms that attempt to guarantee QoS of admitted traffic, either over-provision for admitted services or use effective bandwidth estimation algorithms reliant on strict traffic model assumptions to control QoS of traffic. The former can lead to an under-utilisation of available bandwidth, while the latter one is unable to accurately estimate the effective bandwidth of aggregated traffic flows for non-compliant traffic flows, leading to a sub-optimal utilisation of available resources.

Based on the literature reviewed within this chapter and the issues we have identified, we state the following hypothesis as a motivation of our work:

A purely empirical approach to the estimation of effective bandwidth of a set of aggregated traffic flows can provide the degree of accuracy required to facilitate effective traffic engineering processes such as QoS-aware network planning and admission control.

The remainder of this thesis is devoted to evaluation of the hypothesis through the proposition and evaluation of a purely empirical effective bandwidth estimation algorithm. The algorithm is evaluated within the two traffic engineering scenarios: QoS-aware network planning and admission control. We evaluate efficiency of the approach with respect to both estimation of effective bandwidth of aggregated traffic flows, and cost efficiency within the two processes. We demonstrate that, due to the accuracy of the purely empirical effective bandwidth estimation algorithm, a relatively low cost solution can be developed to establish appropriate input to a QoS aware network planning process. We also demonstrate that a revenue optimisation strategy can be developed for an IPTV admission control algorithm, utilising the purely empirical effective bandwidth estimation algorithm as a method of guaranteeing QoS of admitted traffic while maximising bandwidth utilisation.

Chapter 3

An Empirical Effective Bandwidth Estimation Algorithm

Effective bandwidth is defined as the minimum amount of bandwidth required by a traffic flow to maintain a specified Quality of Service (QoS) related target (Kelly, 1996). There are many approaches proposed to estimate the effective bandwidth of a traffic flow, however the majority of these approaches rely on the formal specification of a traffic model and depend on the traffic within the network being well behaved[1]. We contend that such approaches are unsuitable for use within a communications network, as traffic can be quite difficult to model over varying time-scales, this is backed up by evidence of self-similarity within traffic (Leland *et al.*, 1994).

This chapter specifies and evaluates a purely empirical approach for estimating the effective bandwidth of aggregated traffic flows. We believe such an approach is suitable for use within a communications network as it can operate independently of traffic model assumptions. The approach is evaluated against two estimation approaches reliant on traffic model assumptions. We evaluate these approaches for two common forms of traffic within a communications network; elastic and streaming traffic. We investigate the performance of each algorithm in estimating the effective bandwidth of the traffic flows within a number of scenarios addressing both traffic flow aggregation and QoS target variation. Our experiments focus particularly on estimation of effective bandwidth with regards to QoS targets on packet delay, however the approach can be readily adopted for both loss and delay targets. We proceed by proposing suitable

[1] Well behaved traffic flows are traffic flows that conform to given traffic model assumptions.

operational QoS targets on packet delay for aggregated elastic and streaming traffic based on requirements of the applications that produce this traffic.

This chapter is structured as follows: §3.1 discusses factors that can affect the effective bandwidth of a traffic flow within a communications network. §3.2 presents the proposed empirical approach taken for effective bandwidth estimation. §3.3 presents two traffic model based approaches of effective bandwidth estimation and details an experimental scenario used to evaluate the performance of the three approaches. Finally in §3.4 we summarise the contribution of this chapter and offer some concluding remarks.

3.1 Considerations of Effective Bandwidth Estimation

This section discusses in more detail the factors that can influence the effective bandwidth of a traffic flow. It highlights three contributing factors: traffic characteristics, QoS targets imposed and traffic aggregation.

- **Traffic characteristics**

  For traffic model based effective bandwidth estimation approaches, such as the approaches discussed in section 2.2.1, when the traffic is well behaved the estimation of effective bandwidth can be approximated with reasonable accuracy. However, it is quite difficult to take into account the different activity levels of traffic for different time scales ranging from milliseconds to minutes and hours (Leland *et al.*, 1994). Therefore, a vital requirement for the estimation of effective bandwidth is the ability to estimate this value independent of the traffic characteristics. This will ensure a robust appraoch that can deal with any type of traffic traversing the network, including traffic resulting from the introduction of new applications.

- **Sensitivity to QoS targets**

  Effective bandwidth of traffic is directly related to the stated QoS target being imposed on that traffic. For example, a traffic flow with an absolute zero tolerance to packet delay would require bandwidth to be reserved across the network at the peak throughput of the traffic flow, for it to experience no significant queuing delay across the network. The delay experienced through transmission over links or propagation through network cards is not taken into account. In this case the effective bandwidth of the traffic flow would be equal to the peak throughput. However, QoS targets are rarely as stringent as this, allowing a certain amount of packet delay within the traffic flow. As a result the effective bandwidth of the traffic flow reduces as the QoS targets become less stringent. This behaviour must be reflected within an effective bandwidth estimation algorithm.

- **Effective bandwidth of aggregated traffic**

  Statistical multiplexing is a form of bandwidth sharing between flows of traffic aggregated at a common point on the network. There exists a non-linear relationship between the effective bandwidth of a traffic flow prior to and after the aggregation point (Botvich & Duffield, 1995). The recognition of this behaviour in the estimation of effective bandwidth is crucial, as the aggregation of traffic at ingress points on the network is common practice.

3.2 Empirical Effective Bandwidth Estimation Framework

We intend to demonstrate that a purely empirical estimation of effective bandwidth is the most appropriate approach to measuring effective bandwidth of aggregated traffic flows. The approach is based on the analysis of a recorded packet trace collected from a point in the network. Our approach focuses solely on estimating the effective bandwidth of a traffic trace for specified QoS targets on packet delay. It is important to note that the approach presented here is a per hop estimation of effective bandwidth. The approach does not take into consideration end to end packet delay across the network, including propagation delay and link delay.

Recent work has been published on an approach for empirically measuring the effective bandwidth of a packet trace through the simulation of a FIFO queue (Liu & Baras, 2004). The approach proposed here is similar in nature, however we have explicitly defined the FIFO queue model we wish to implement, where as Liu & Baras (2004) have not. We have also specified the specific binary search algorithm used to find the effective bandwidth value. A difference between the two approaches is also with respect to the duration of the packet trace used for analysis. We choose an appropriate packet trace duration to suit the particular scenario and performance optimisation problem being developed to use the effective bandwidth estimation, on the other hand Liu & Baras (2004) define a method of optimising the duration of the packet trace size, used for analysis, to avoid over-estimation. We believe, for the types of scenarios we consider, this optimisation step is unnecessary. We evaluate the decision of selecting a packet trace size later in section 3.3.4.4.

A typical example of a QoS delay target is (0.04 s, 0.001), which means that only 0.1% of traffic is allowed to be delayed more than 40 ms. As the effective bandwidth depends on the QoS target, for different QoS targets, effective bandwidth estimations could be different. Suppose the QoS delay target is fixed and includes $delay_{max}$ the maximum delay and p_{delay} the proportion of traffic which can exhibit delay more than $delay_{max}$. We define effective bandwidth R_{eff} of a traffic flow for the QoS delay target $(delay_{max}, p_{delay})$ as a minimal link rate such that if we simulate a FIFO queue (with an unlimited buffer and assumed initially empty) the proportion of traffic which will exhibit delay more than $delay_{max}$ will be less than p_{delay}.

To estimate the effective bandwidth of a particular traffic flow, we take a recorded packet trace of that flow from the network. We observe that if we simulate a FIFO queue with the same inputted packet trace $\{T_M\}$ for different queue service rates $R_1 < R_2$ and estimate the proportions p_1 and p_2 of traffic delayed more than $delay_{max}$ for different rates respectively, then $p_1 > p_2$. This means that the proportion of traffic, p, delayed more than $delay_{max}$ is a monotonically decreasing function of service rate R. Using this observation we define, a simple binary search algorithm for a recorded packet trace to find the minimal value of a queue rate such that the proportion of traffic delayed more than $delay_{max}$ is less than p_{delay}.

3.2.1 Algorithm for Estimating Proportion of Violating Traffic using the FIFO Queue

There are two approaches commonly used to model a FIFO queue, the packet model or the continuous model. As depicted in Fig. 3.1, the packet level FIFO queue models the processing of each packet as a whole, where as a continuous FIFO queue models the processing of packets as a continuous bit stream. The latter approach distinguishes between total and partially delayed packets, including only the volume of traffic delayed in the calculation. The former approach will include the volume of a complete packet

Notation	Description
$T(x_i, t_i)$	A packet within a trace list where packet size x_i and corresponding arrival times t_i.
$\{T_M\}$	The set of all packets contained within trace T_M.
M	The total number of packets in trace T_M.
$delay_{max}$	Maximum allowable packet delay target in seconds.
p_{delay}	Target proportion of traffic allowed to violate $delay_{max}$.
R	The service rate of the FIFO queue.
R_{mean}, R_{peak}	Measured mean and peak throughout rates of packet trace T_M.
p	Corresponding proportion of violating traffic for service rate R, packet trace T_M and delay target $delay_{max}$.
β	Identifies a margin of accuracy used to find the effective bandwidth value for a particular QoS violation target as a proportion of the QoS delay target p_{delay}.
e	Identifies the actual margin of accuracy to be used relative to p_{delay}.
R_{eff}	FIFO queue service rate that meets QoS requirements on proportion of traffic violating the specified packet delay target. We use this value as a measure of Effective Bandwidth.
δ_{max}	Maximum queue volume, before traffic is delayed greater than $delay_{max}$.
δ_{time}	Time to service the queue at service rate R, from the time of packet arrival.
δ_{vol}	Current volume of the FIFO queue buffer.
$TOTAL_{vol}$	Total volume of traffic that has been processed through the FIFO queue.
$DELAY_{vol}$	Volume of traffic delayed greater than the target $delay_{max}$.

Table 3.1: Notation for the estimation of effective bandwidth

into the calculation, even if only partially delayed. With the inclusion of total packet size for partially delayed packets in the calculation of traffic violations, it is safe to assume that a packet model would result in a more conservative estimation of effective bandwidth. For the purpose of this work, we implement a continuous FIFO queue model for a more fine grained estimation of effective bandwidth to be guaranteed.

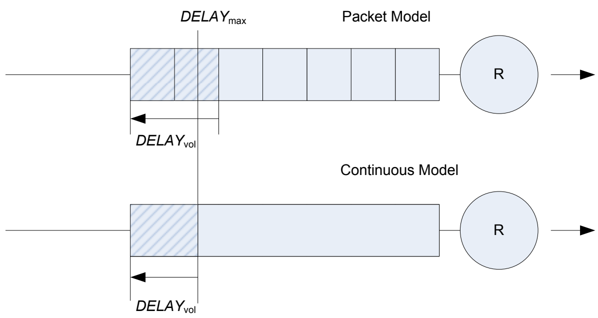

Figure 3.1: FIFO Queue packet model versus the continuous model

In Alg. 3.1 we define our continuous FIFO queue model for use in the empirical estimation of effective bandwidth. This algorithm is used to calculate the proportion of violating traffic for a particular queue service rate. Table 3.1 summarises the FIFO queue algorithm notation. Each packet in the trace consists of a pair of attributes that specify the packet size, denoted x_i in bits and packet arrival time denoted t_i in seconds. Let δ_{max} denote the maximum allowable volume of the queue buffer in bits before traffic experiences delay greater than $delay_{max}$. Let δ_{vol} denote the current volume of the queue buffer. Let $TOTAL_{vol}$ denote the total volume of traffic that has passed through the queue. Let $DELAY_{vol}$ denote the total volume of traffic that has exceeded the allowable bound of δ_{max}. Finally, p denotes the proportion of traffic delayed in respect to total traffic processed.

The algorithm assumes an infinite queue buffer, which is initially empty. The justification for using an infinite buffer is to ensure no packets are lost during the processing of the packet trace through the FIFO queue. To consider QoS targets of packet loss, a limit on the queue buffer would be imposed. The algorithm is passed a specified service rate R to process the packet trace, a specified maximum delay target on traffic $delay_{max}$, and a packet trace $\{T_M\}$ as input. Once the queue is initialised, the algorithm sets a specified target queue volume δ_{max} by multiplying the specified service

Algorithm 3.1: FIFO queue algorithm for estimation of violating traffic

Input: $delay_{max}$, R, $\{T_M\}$
Output: p

Set $\delta_{vol} = 0$;
Set $\delta_{time} = 0$;
Set $TOTAL_{vol} = 0$;
Set $DELAY_{vol} = 0$;

Set $\delta_{max} = delay_{max}R$;

forall $T(x_i, t_i)$ *in* T_M **do**

$TOTAL_{vol} = TOTAL_{vol} + x_i$;

//IF Queue is empty
if $t_i \geq \delta_{time}$ **then**
$\delta_{time} = t_i + \frac{x_i}{R}$;

//Else packet is waiting
elseif $t_i < \delta_{time}$ **then**
$\delta_{vol} = (\delta_{time} - t_i)R + x_i$;
$\delta_{time} = \delta_{time} + \frac{x_i}{R}$;

//Traffic in violation
if $\delta_{vol} > \delta_{max}$ **then**
$DELAY_{vol} = DELAY_{vol} + (\delta_{vol} - \delta_{max})$;

end
Set $p = \frac{DELAY_{vol}}{TOTAL_{vol}}$;

return p;

rate R and delay target $delay_{max}$ to calculate the maximum volume the queue can be before traffic is delayed greater than this specified target. If the volume of the queue δ_{vol} exceeds the maximum queue limit, all traffic beyond this limit will experience delay greater than the specified target.

The algorithm processes the packet trace as follows: For each packet, the packet arrival time t_i is compared to the time set after the queue has been emptied, following the previous packet arrival δ_{time}. If the packet arrival time is greater or equal to δ_{time}, then the queue is empty and the packet is processed. The packet is processed by updating the queue time by the time it takes the queue to process the packet at the specified queue service rate. If the packet arrival time is less than δ_{time}, this means the packet must wait to be processed. To calculate the waiting time, we must calculate the difference between the packet arrival time and the current queue time. The queue must process traffic for this duration before processing the arrived packet. At this point, we store the volume of traffic within the queue by adding the volume of traffic ahead of the packet plus the packet itself. If at this stage the queue volume δ_{vol} is greater than the maximum allowable queue volume δ_{max}, the difference is recorded as the volume of traffic in breach of the delay target.

As the algorithm proceeds, the total volume of traffic delayed greater than the specified delay target accumulates in $DELAY_{vol}$. When all packets have been processed through the queue, the algorithm will return the proportion of delayed traffic $DELAY_{vol}$ over the total volume of traffic processed, $TOTAL_{vol}$.

3.2.2 Effective Bandwidth Binary Search Algorithm

The effective bandwidth estimation algorithm, as depicted in algorithm 3.2, controls the FIFO queue service rate to find a suitable service rate R where the proportion of violating traffic p is equal to the specified violation target p_{delay}. However, as p is a real number, comparisons of equality are difficult, therefore we implement an error region, denoted β. This error region enables us to control the precision of the algorithm in estimating the effective bandwidth of a recorded packet trace.

The algorithm , as outlined in Alg. 3.2 takes as parameters, a specified traffic delay target, denoted $delay_{max}$, a specified violation target (denoted p_{delay}), a packet trace (denoted $\{T_M\}$), and an error control variable (denoted β). p denotes the corresponding proportion of violating traffic for service rate R using the specified traffic delay target.

Algorithm 3.2: Estimation of effective bandwidth using a binary search algorithm

Input: $delay_{max}$, p_{delay}, $\{T_M\}$, β
Output: R_{eff}

Set $e = p_{delay}\beta$;
Set $R_{low} = calcMean\left(\{T_M\}\right)$;
Set $R_{high} = calcPeak\left(\{T_M\}\right)$;
Set $R_{mid} = R_{low} + \frac{R_{high} - R_{low}}{2}$;
Set $p_{mid} = calcViolations\left(delay_{max}, R_{mid}, \{T_M\}\right)$;
while $(p_{mid} = (p_{delay} \pm e)) = false$ **do**
    if $(p_{mid} < p_{delay})$ **then**
        $R_{high} = R_{mid}$;
    else
        $R_{low} = R_{mid}$;
    Set $R_{mid} = R_{low} + \frac{R_{high} - R_{low}}{2}$;
    Set $p_{mid} = calcViolations\left(delay_{max}, R_{mid}, \{T_M\}\right)$;
end while
$R_{eff} = R_{mid}$;
return R_{eff};

The objective of the algorithm is to find a service rate R where its corresponding proportion of violating traffic p is equal to p_{delay} ($\pm$ the error region β). The error region is calculated as $\pm$ a specified percentage of p_{delay} dictated by the β attribute. This β parameter is required as the values of p and p_{delay} being compared are real numbers over a continuous search space. These numbers cannot be compared directly and are therefore compared based on a margin of error specified by the β parameter. Therefore, the higher the β parameter, the wider the region of comparison is between the values.

3.3 Evaluation of the Empirical Effective Bandwidth Estimation Algorithm

We now provide an in-depth analysis and evaluation of the empirical effective bandwidth estimation algorithm. We first analyse the time and space complexity of the algorithm, discussing methods of improving performance and scalability in different traffic conditions. We then discuss the settings of our network scenario and perform a number of sensitivity analysis experiments on the input parameters of the algorithm. We finally evaluate the performance of the algorithm for estimating effective bandwidth of aggregated traffic flows in comparison to two traffic model based effective bandwidth estimation approaches.

3.3.1 Time and Space Complexity

We first focus on the time complexity of the proposed algorithm. As the algorithm relies on the simulation of a FIFO queue model to estimate the proportion of QoS violations that a packet trace incurs for a particular queue service rate, each packet within the packet trace must be processed in succession. This operation happens in time O(N), where N is the number of packets within the packet trace. As the algorithm uses a binary search algorithm to choose appropriate service rates dependent on the associated QoS violations experienced, the algorithm must repeat the previous operation in time O(log$M$), where M is the search space of possible QoS violations. M is dependent on both the QoS violation target, p_{delay} and the error resolution parameter β. This error region, calculated as $p_{delay} * \beta$, is used by the algorithm to evaluate whether an appropriate QoS violation target has been found. The algorithm evaluates $\pm$ this value of the calculated p against the target p_delay. Therefore we can calculate M as $\frac{1}{p_{delay}\beta}$. As the algorithm employs a binary search strategy to locate the QoS violation target and corresponding queue service rate, the theoretical number of iterations can be found as follows:

$$\log_2(\frac{1}{p_{delay}\beta}) \tag{3.1}$$

The algorithm therefore runs in O(Nlog$M$) time. The smaller the QoS violation target is, and the smaller the error region will be, the larger the search space will be. The performance of the algorithm can therefore be improved by two means; firstly by reducing the number of packets within the collected packet trace to reduce N, and secondly increasing either the QoS violation target or the error resolution parameter β. The first can be achieved by collecting a shorter packet trace or by collecting a trace at a lower resolution of time (e.g. bits per millisecond). In the second case, however, the QoS violation targets tend to be predetermined by the types of traffic and the higher level SLAs and QoS guarantees offered by the network operator. Therefore the controlling parameter to be used is β.

To evaluate the time complexity of the proposed algorithm we perform the following experiment. We replay a packet trace through the algorithm and vary the β parameter and with a fixed QoS violation target, p_{delay}, of 0.001. We measure the number of iterations of the algorithm for a number of different β values. The results are then plotted against the theoretical binary search function. The packet traces for this experiment were taken from the MOME data-set MOME (2008). Two packet traces of a duration of 6.35 hours were processed through the algorithm at 5 minute segments. Each segment is processed through the algorithm and the number of iterations are recorded. Fig 3.2 depicts both the experimental results collected from $\tilde{1}50$ algorithm iterations using different β values and the theoretical binary search of an equivalent search space; as can be seen the algorithm performs in line with the expected theoretical equivalent.

With regards to the space complexity of the algorithm, a packet trace is loaded into memory in its entirety once before the algorithm is executed. Therefore the space complexity of the algorithm is O(N), where N is the number of packets within the trace. Efficiency can therefore be increased by reducing the size of the packet trace or by increasing the resolution at which the trace is collected.

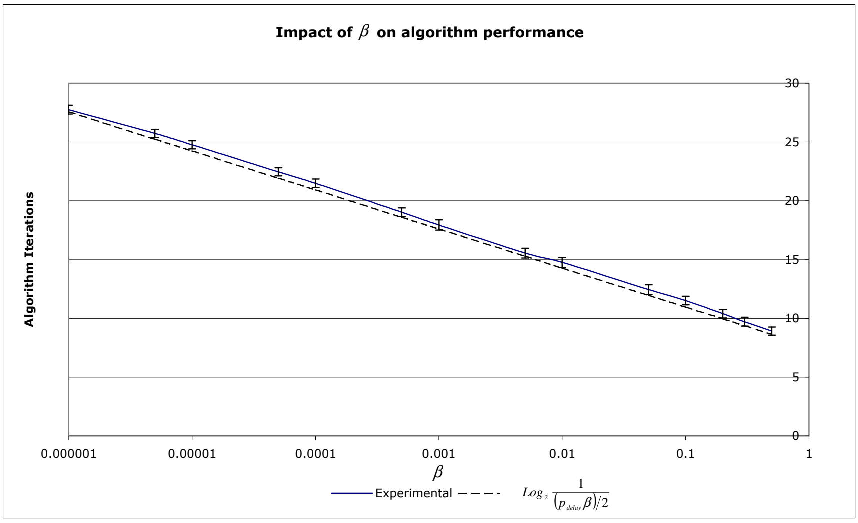

Figure 3.2: Affect varying β has on algorithm performance.

3.3.2 Simulation Settings

We now discuss the characteristics of the traffic and proposed operational network scenario we will use throughout the experimental evaluation process.

3.3.2.1 Traffic Characteristics

We simulate both elastic and streaming traffic conditions to demonstrate the ability of our empirical approach to estimate effective bandwidth of such traffic types. To simulate elastic traffic, we collected an IP packet trace of all HTTP traffic entering the corporate network at our facility. The facility has a 250Mbps connection to the Internet over which all external traffic enters the network. The packet trace collected contains a total of 85 end point hosts within the ArcLabs (2008) network, connecting to up to 780 different external servers over the duration of the trace.

Within the facility there are about 100 staff members plus a number of internal software companies. The staff profiles consist of both researchers and software developers at a ratio of 60/ 40 with a limited number of non technical personnel covering

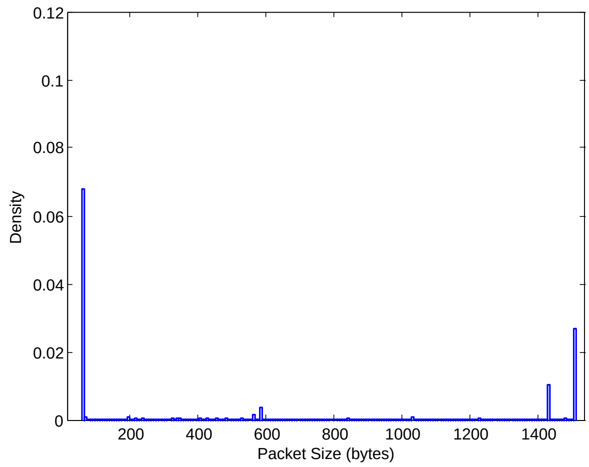

Figure 3.3: Distribution of packet sizes for the HTTP packet trace.

administrative and operational duties. The trace was collected using Ethereal with the following filter:

```
src port 80 http
```

The packet trace was collected over the period of 4pm and 5pm on February 11^{th} 2008; during this period the mean throughput was 4.4Mbps. Once the packet trace was collected, the packet size and arrival time was extracted from the each packet header and stored. Figs. 3.3 and 3.4 depict the properties of the collected elastic packet trace. Fig. 3.3 depicts the distribution of packet sizes within the traffic. The distribution shows that the traffic contains a bimodal distribution, where a large proportion of packets are either very small ($\tilde{6}0$ bytes) or very large ($\tilde{1}500$ bytes). This has also been observed in a recent Internet packet size distribution studie (Sinha *et al.*, 2007). This is typical for HTTP traffic, as TCP control signalling such as ACK would consume very little bandwidth while data traffic such as images would consume a larger volume of traffic.

We also wish to model streaming traffic within the network to evaluate the behaviour of our algorithm within this context. We use a video frame trace detailed in Fitzek & Reisslein (2001) to simulate streaming traffic within the network. The frame trace is taken from the film Jurassic Park over a duration of 1 hour. Fig. 3.5 depicts the frame size distribution of the frame trace being used. The video is simulated as being

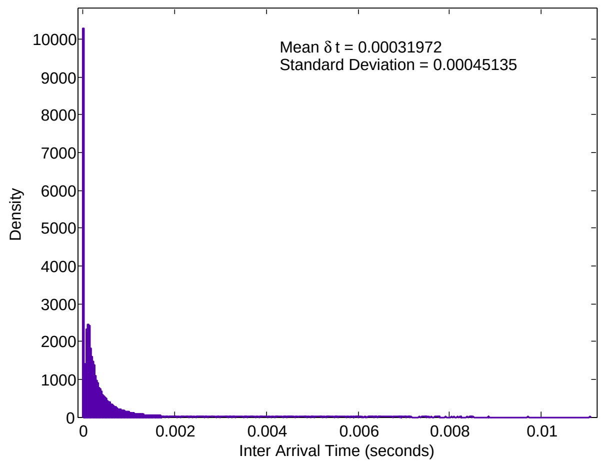

Figure 3.4: Distribution of packet inter-arrival times for the HTTP packet trace.

streamed across the network at a constant frame rate of 24 frames per second. Fitzek & Reisslein (2001) have demonstrated that this frame trace has self-similar properties, deeming it very difficult to accurately model its characteristics. We use this frame trace for this reason, with the intention of demonstrating the effectiveness of our purely empirical approach of measuring effective bandwidth in comparison to traffic model reliant approaches.

3.3.2.2 Simulated Network Topology

The following network topology is used to evaluate the performance of the discussed algorithms. The bottleneck topology depicted in Fig. 3.6 is configured and modeled in the OPNET$^{\text{TM}}$Modeler (OPNET, 2008). Traffic is generated at the attached source nodes using the discussed packet traces, aggregated onto an OC3 155 Mbps link and forwarded to the ingress node of a core network. Of this 155 Mbps, we assume 10% is reserved for emergency scenarios, so we assume 140 Mbps of bandwidth is available. A measurement device is set up on the aggregation link where packet traces of the generated traffic are collected for analysis. As the same packet trace is being used to model multiple traffic sources, OPNET has been set up to begin using the packet trace at each source at a random point within the trace which wraps around for the duration of the simulation. This is to ensure a diverse set of traffic characteristics are

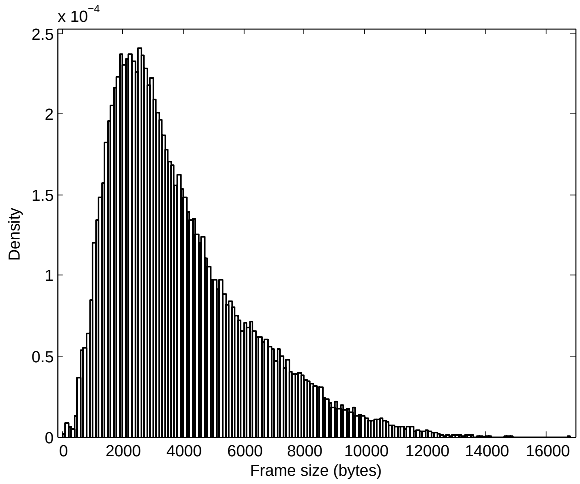

Figure 3.5: Distribution of frame sizes for streaming video traffic.

simulated throughout the running of the simulation. Packet traces are collected at the measurement point of equal duration $T_{max} = 5$ mins. We discuss the significance of this interval choice later in section 3.3.4.4.

3.3.3 Comparison between purely empirical and traffic model reliant estimation of effective bandwidth

We define the following comparative analysis strategies with the intent of demonstrating the advantages of estimating the effective bandwidth of a packet trace empirically over approaches reliant on traffic model assumptions. Our intention is to demonstrate the effect their inability to recognise the effect statistical multiplexing has on the estimated effective bandwidth of varying levels of aggregated traffic flows. We evaluate two approaches; the Gaussian estimation (Guérin *et al.*, 1991) and the Roberts estimation (Roberts *et al.*, 1996).

3.3.3.1 Gaussian Estimation

For the Gaussian estimation technique for elastic traffic it is assumed that traffic arriving at the point of measurement follows a Gaussian distribution. Using this assumption Guérin *et al.* (1991) defines the following formula to estimate the effective bandwidth,

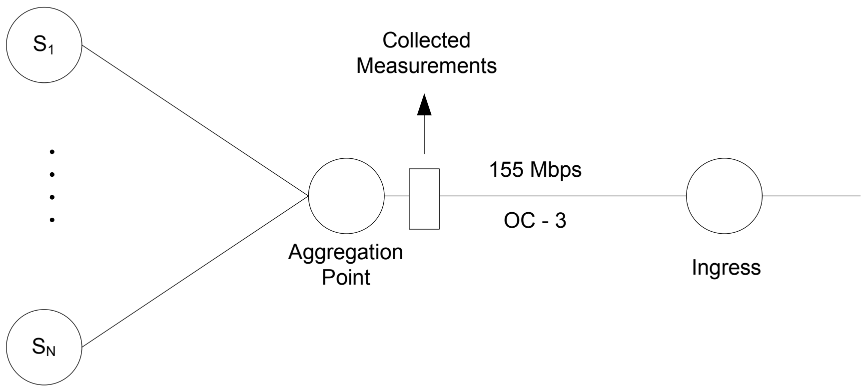

Figure 3.6: Simulated Network Topology.

B_{eff} , of a packet trace based on the mean throughput m, and the standard deviation of the packet arrival rate of arriving traffic σ, for a specified packet loss ratio ϵ:

$$B_{eff} = m + \sigma\sqrt{-2\ln\epsilon - \ln 2\pi} \tag{3.2}$$

3.3.3.2 Roberts Estimation

The Roberts estimation technique is used to estimate the effective bandwidth of streaming traffic. It estimates the maximum link utilisation threshold ρ_{max} required to maintain set QoS targets on packet delay (Roberts *et al.*, 1996). The approach models traffic arriving on a link as an $N \cdot D/D/1 - \infty$ queueing system, with N homogeneous flows, each sending packets of a mean expected packet size $E[B]$ and inter-arrival time $E[A]$. The rate of a particular flow, C_f is calculated as follows:

$$C_f = \frac{E[B]}{E[A]}. \tag{3.3}$$

The packet delay distribution of the queue system is modelled by:

$$\begin{gathered} x = \frac{t \cdot B_{TOT}}{E[B]} \qquad \textit{and} \qquad \rho = \frac{N \cdot E[B]}{C_f} \\ P(W \leq t) = 1 - e^{-2x(\frac{x}{N}+1-\rho)} \end{gathered} \tag{3.4}$$

where B_{TOT} is the total available bandwidth on the link.

The objective of the algorithm is to find the maximum link utilisation for which the specified QoS target on packet delay is satisfied, where W_{max} is the target maximum packet delay time and p_W is the proportion of traffic allowed to violate this target. The maximum link utilisation ρ_{max} is computed by:

$$\rho_{max} = \max_{\rho} \rho : P(W > W_{max}) \leq p_W \tag{3.5}$$

3.3.4 Choice of algorithm settings

We now discuss the set of experiments we perform to compare the algorithms outlined in section 3.3.3 against our empirical effective bandwidth estimation algorithm. We first wish to demonstrate the operation of our algorithm in finding the effective bandwidth of a packet trace. Following this, we discuss the choice of measurement resolutions in the calculation of the peak rate of a packet trace. As the binary search algorithm uses the peak rate of the collected packet trace as as its upper limit, we must ensure this measurement is being attained in an appropriate manner. We evaluate the effect of varying the β value has on the performance of the algorithm. We then go on to discuss the decision of choosing an appropriate packet trace duration T_{max}. Following these set of experiments specific to the algorithm itself, we evaluate the algorithms effectiveness at measuring the effective bandwidth of aggregated traffic flows, in comparison to the proposed approaches reliant on traffic model assumptions. Finally, we evaluate the affect varying QoS violation targets have on the estimation of effective bandwidth through our algorithm.

3.3.4.1 Demonstration of empirical effective bandwidth estimation

As can be seen in Fig. 3.7 the HTTP packet trace packet trace has been processed through the FIFO queue at a service rate of 516,847 bps. The maximum queue size is calculated to be 103,369 bits, based on an outlined QoS delay target of 0.04 sec. For this particular scenario, we set a QoS violation target of 0.001 and a β value of 0.01. If the queue size exceeds the maximum allowable queue size, the violating volume of traffic will be delayed greater than our QoS delay target.

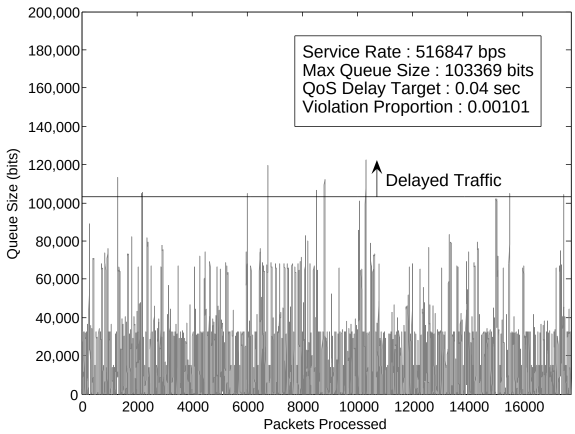

Figure 3.7: Volume of traffic violating the QoS target on packet delay for a particular service rate.

The FIFO queue algorithm keeps track of the amount of traffic delayed greater than this limit. Once the packet trace has been processed, the fraction of traffic violating the QoS delay target is calculated by dividing the delayed volume of traffic by the total queue volume accumulated. In this case, our algorithm detects that the violation proportion is within acceptable limits of the required QoS violation target for this packet trace and will return 516,847 bps as the effective bandwidth estimation.

3.3.4.2 Resolution of peak rate measurements

The measurement resolution is used to estimate the peak rate of the collected packet trace. We perform an experiment of calculating the peak and mean rate of the HTTP packet trace using different measurement resolutions ranging from 0.0001 seconds to 10 seconds. The effective bandwidth binary search algorithm assumes that the effective bandwidth value we are looking for is between the peak rate and mean rate of the packet trace being processed. However, if a large measurement interval is being used to estimate the peak throughput of the packet trace, this assumption may not be valid, as the effective bandwidth may be higher than that of the measured peak rate. We must ensure that a reasonable level of accuracy is being used in the estimation of the peak rate.

From Fig. 3.8, we see that the measured peak rate reduces as the measurement in-

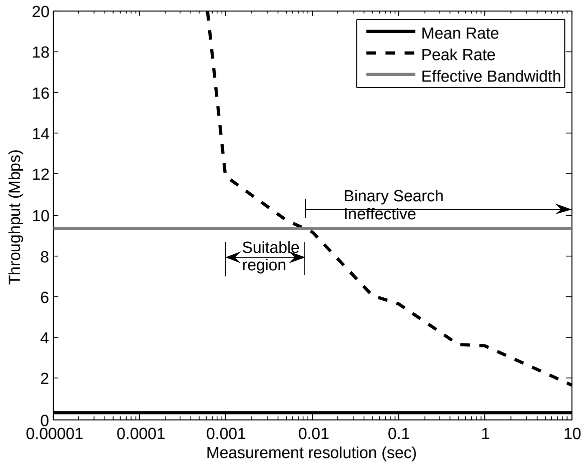

Figure 3.8: Variation on peak rate measurements with respect to measurement resolution.

terval increases. This figure also plots the estimated effective bandwidth of the packet trace using appropriate QoS targets on packet delay. Within the figure we have marked an ineffective search region for the binary search algorithm as the effective bandwidth is higher than the peak measurement. Within such a situation, an appropriate effective bandwidth value would not be found by the binary search algorithm. For the experiments within this thesis using the packet trace stated, we use a measurement interval of 0.001 to measure the peak rate. We believe this to be accurate enough for our purposes, ensuring the effective bandwidth will be within the range of peak and mean, while also ensuring that not too high a peak value is being used to within the effective bandwidth search algorithm.

3.3.4.3 Error Region

We now analyse how the algorithm's performance is affected by varying the β value used to discover the appropriate proportion of violating traffic, using the proposed binary search in Alg. 3.2. The β value controls the algorithm's decision of whether an appropriate p value matching the QoS packet violation target of p_{delay} has been discovered, as depicted in Fig. 3.9. As the algorithm is searching a continuous search space, the larger the β value, the more margin of error is acceptable in testing equivalence between the two values. This margin or error determines the level of accuracy required

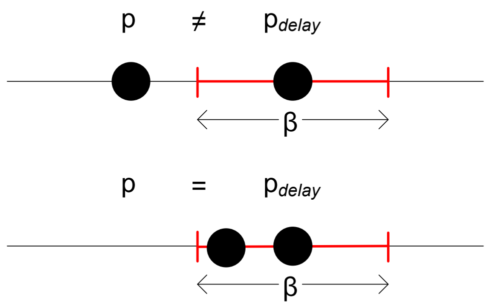

Figure 3.9: Use of the β value within the binary search algorithm.

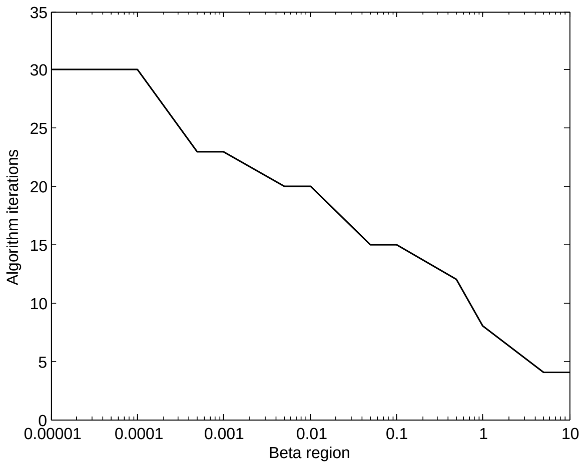

Figure 3.10: Effect of β has on algorithm iterations.

for finding the effective bandwidth, with a trade off against algorithm performance (with respect to number of iterations) for higher precision searches. We execute the algorithm on the HTTP packet trace supplied.

As can be seen in Fig. 3.10, for a very small β value, there are a high number of algorithm iterations. The number of iterations reduces as the β value expands, however this results in a reduction in precision of the effective bandwidth estimation. We wish to specify an appropriate β value suitable for the experiment we intend to carry out. Within the experiments, we choose a β value of 0.01. With this value, we can guarantee the effective bandwidth value found will have the QoS violation of $\pm$ 0.01 of the target violation for HTTP packet traces collected from the network.

3.3.4.4 Packet Trace Duration

We now discuss the choice of T_{max}, dictating the duration of the packet trace collected from the network and analysed to estimate effective bandwidth. For an hour long HTTP packet trace, we compared the effective bandwidth estimations over a number of T_{max} intervals. As can be seen from Fig. 3.11, at $T_{max} = 3600sec$, the algorithm returns the highest effective bandwidth value over the entire period. This estimation is the effective bandwidth required to maintain the QoS packet delay target for the packet trace over the complete hour. However the resultant effective bandwidth estimation may not reveal the variations in effective bandwidth throughout that particular hour and therefore may not be suitable for use in realtime traffic optimisation strategies such as admission control or queue management, which deal with decisions over much shorter time scales.

For packet traces collected with $T_{max} = 300sec$, we notice a certain degree of variation in required effective bandwidth over the hour. These measurements can be more suitable for near-realtime traffic optimisation strategies such as admission control where bandwidth must be managed over seconds and minutes, instead of hours. Again, the effect is magnified when the packet traces are collected over $T_{max} = 60sec$. With the monitoring of smaller packet traces, we can build a more accurate picture of the variation in effective bandwidth over a long period. We will discuss the usage of short term effective bandwidth estimation in chapters 4 and 5.

3.3.5 Effective Bandwidth of Aggregated Traffic

Multiplexing is a common technique employed for sharing bandwidth between traffic flows at a point of aggregation (Maglaris *et al.*, 1988). However, within the aggregated traffic flow, the effective bandwidth the individual traffic flows are also affected by this process, as is theoretically demonstrated by Botvich & Duffield (1995) and Simonian & Guibert (1995). The recognition of this effect is crucial to estimating bandwidth requirements of aggregated traffic flows accurately. The objective of this experiment is to evaluate whether the proposed algorithm considers this effect for the estimation of effective bandwidth of aggregated traffic flows.

The experiment is performed separately for an aggregate of elastic and streaming traffic. As these two traffic types have different QoS targets, the effective bandwidth

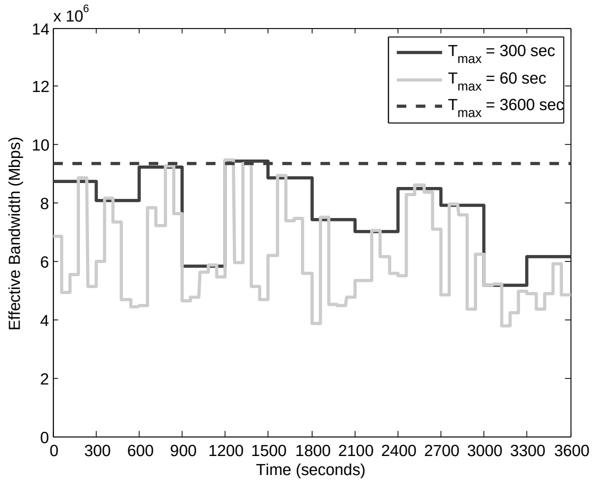

Figure 3.11: Effective bandwidth estimations for various packet trace durations.

per traffic aggregate must be measured separately. The QoS packet delay targets are set to (0.02 sec, 0.0001) for streaming video traffic and (0.04 sec, 0.001) for elastic HTTP traffic. The experiment progresses by measuring the effective bandwidth of elastic traffic at the point of aggregation using both the empirical algorithm and the Gaussian estimation, as an increasing number of flows are aggregated. The experiment is repeated for streaming traffic flows to evaluate the suitability of the Roberts estimation of capturing this affect for a different traffic type, in comparison to the empirical algorithm.

An increasing number of flows (1 - 64) are streamed from the source nodes, through the aggregation router and into the ingress point of the network. Packet traces are collected over a period $T_{max} = 300sec$. The collected packet traces are then processed to estimate an appropriate effective bandwidth value to suit the outlined QoS packet delay targets, by each of the two traffic aggregates. For the benefit of result comparisons between elastic and streaming traffic, we normalise the effective bandwidth values estimated against the measured mean throughput of each collected packet trace. The associated settings for the effective bandwidth estimation approaches reliant on traffic model assumptions are outlined in Table 3.2.

The Gaussian estimation depends on measurements of mean throughput and standard deviation of the packet arrival rate of the collected packet trace. For this experi-

Table 3.2: Traffic model settings for the estimation of effective bandwidth.

Algorithm	Attribute	Settings
Gaussian	ϵ	0.001
Roberts	$E[A]$	0.0043 seconds
	$E[B]$	9954.38 bits
	t	0.05 sec
	p_W	0.9999

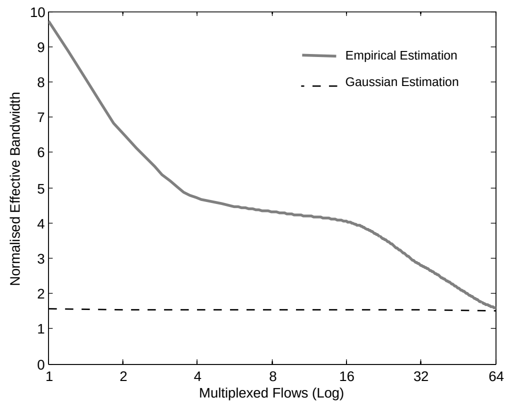

Figure 3.12: Empirical versus Gaussian estimation of effective bandwidth of aggregated traffic flows (Elastic).

ment we set ϵ (the packet loss ratio) to 0.001 to match the QoS violation target set for elastic traffic. For the Roberts estimation algorithm we used the following settings to represent streaming traffic, which have been calculated from the outlined packet trace. Streaming flows are assumed to have a packet arrival time of $E[A] = 0.0131$ sec, and a packet size of $E[B] = 9,978.3$ bits. These values have been estimated from performing an analysis of the packet traces generated from streaming the associated video frame trace through the simulated network. The link has a maximum throughput of $c(l) =$ 140 Mbps. The QoS target on packet delay for streaming traffic is set to $t = 0.02$ seconds, with a probability of violation $p = 0.0001$. Based on these settings, the maximum link threshold is estimated at 0.8646.

Figs. 3.12 and 3.13 depict the normalised effective bandwidth per level of aggregated flows, for both elastic and streaming traffic respectively. For both the elastic and streaming traffic, we observe the normalised effective bandwidth per traffic flow reduces

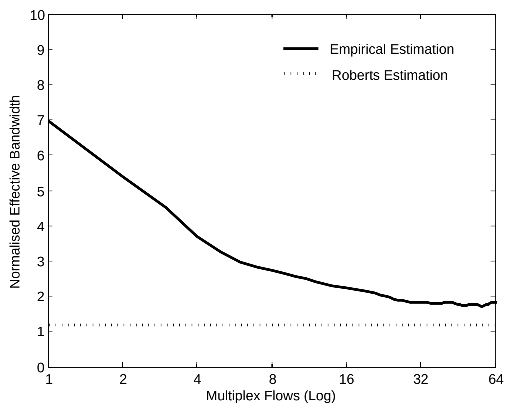

Figure 3.13: Empirical versus Roberts estimation of effective bandwidth of aggregated traffic flows (Streaming).

as the number of aggregated traffic flows increase. However, for both approaches reliant on traffic model assumptions, there is no change in the normalised effective bandwidth per traffic flow. The reason for this is that these approaches assume high levels of aggregation with the traffic. For example, for the Gaussian estimation, the algorithm is based on the assumption that distribution of packet arrivals follow a Gaussian distribution. Such an assumption is only valid for very high levels of aggregation as discussed in Guérin *et al.* (1991). Therefore, this algorithm is only positioned to estimate effective bandwidth at high levels of traffic aggregation. Also for the Roberts estimation, the maximum link utilisation threshold is calculated for the maximum number of flows allowed on the link, thus the algorithm assumes a high level of aggregation in the estimation process, and will reserve a static amount of effective bandwidth for each flow, regardless of the level of aggregation.

3.3.6 Variation in QoS Violation Targets

We now study the effect varying QoS violation targets have on the effective bandwidth of aggregated traffic flows using the empirical effective bandwidth algorithm. The objective of this experiment is to demonstrate that varying the QoS violation targets can affect effective bandwidth estimations across levels of aggregation. We evaluate this effect for both elastic and streaming traffic, and offer some advice on typical operational QoS violation target regions suitable for applications that generate these types of traffic.

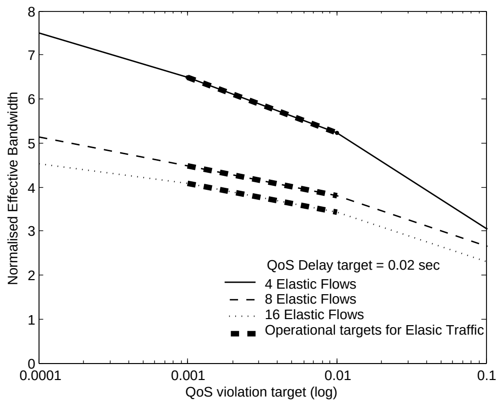

Figure 3.14: Operational QoS violation targets of elastic traffic of varying traffic flow aggregations.

For each simulation run the QoS target set for the proportion of violating traffic is modified from 0.0001 to 0.1. The experiment is repeated for 4, 8 and 16 traffic flows of elastic, then streaming traffic. Figs. 3.14 and 3.15 demonstrates that as the QoS violation target is relaxed, the required amount of effective bandwidth for each aggregate of traffic decreases, for both elastic and non elastic traffic respectively. The main reason for this is, as the QoS targets are relaxed, less bandwidth is required to maintain the QoS targets imposed on the traffic. We notice that the decrease is at a different rate for elastic and streaming traffic. This is due to the characteristics of elastic and streaming traffic, elastic traffic may have weaker QoS targets on packet delay but would have a high degree of bursts capable of violating the QoS targets on packet delay. On the other hand, streaming traffic generally requires strict QoS targets on packet delay, but with relatively stable streaming of packets, bursty behaviour resulting in QoS breaches occurs less often.

To ensure adequate resources are being reserved for the traffic, it is important to choose appropriate QoS violation targets to impose on traffic. Setting the QoS violation target too high can result in an over provisioning of resources for the traffic. On the other hand, too low a setting of the QoS violation target may render the service unusable to the user in times of congestion, as there is an under provisioning of resources, and an excessively high proportion of delayed traffic allowed. Common operational QoS targets for proportion of traffic violations set on elastic traffic range from 0.001 to 0.01. Such

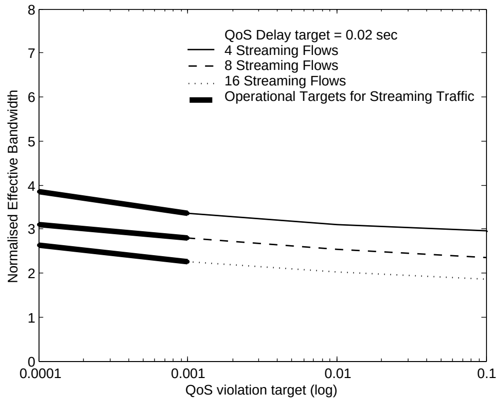

Figure 3.15: Operational QoS violation targets of streaming traffic of varying flow aggregations.

a range with an associated QoS delay target of 0.04 seconds will ensure applications such as HTTP, FTP and email will perform acceptably over the network. These targets must always be set in co-ordination with outlined Service Level Agreements between the consumer and service provider. For the streaming traffic aggregate, we suggest a QoS violation target ranging from between 0.0001 and 0.001 subject to a QoS packet delay target of 0.02 seconds. Fig 3.15 also demonstrates how the effective bandwidth within this region can vary dependent on the level of aggregation within the traffic.

3.3.7 Further analysis of effective bandwidth estimation on various packet traces

We now provide a further analysis of the estimated effective bandwidth values of publicly available packet traces of various sizes. We provide a table with the relevant packet trace details and associated results. The packet traces are taken from two popular IP packet trace data sets, namely the MOME data-set (MOME, 2008) and the CAIDA data-set (CAIDA, 2008). Further details of the associated packet traces can be found in Appendix B.1.

From this table we see that for very low bandwidth traffic, such as the initial two analysed packet traces, the empirically estimated effective bandwidth to mean ratio is very high. On the other hand, the Gaussian formula estimates the required effective bandwidth to mean ratio as much smaller. This is in line with the results displayed in both Figs. 3.12 and 3.13. As the bandwidth of the traffic increases such as in the later three packet traces, we notice that for the empirical effective bandwidth to mean ratio reduces, again in line with previously presented results.

3.4 Summary and Conclusions

In this chapter we presented an empirical approach for estimating the effective bandwidth of traffic that is suitable for use within a communications network. We highlighted the disadvantages of effective bandwidth approaches reliant on strict traffic model assumptions in particular their inability to react to variations in traffic characteristics such as the effect of statistical multiplexing. As modelling the behaviour of traffic can be quite a challenging task, such algorithms can produce highly inaccurate estimations of effective bandwidth as traffic deviates from traffic model assumptions.

In the specification of the empirical effective bandwidth estimation algorithm, a number of control variables that influence its operation were identified and evaluated. These variables are the resolution of measurement used to estimate the peak rate of a packet trace, the error region used to evaluate if an appropriate proportion of traffic violations has been detected, the duration of a packet trace being collected, and the QoS targets. The importance of choosing an appropriate measurement resolution in the estimation of peak throughput of a packet trace was discussed. The proposed effective bandwidth estimation algorithm assumes that the effective bandwidth lies between the peak rate and the mean rate of a collected packet trace. We demonstrated that as the measurement interval increases, the measured of peak rate throughput decreases. This is the result of smoothing over larger measurement intervals. As a binary search algorithm is used to search for the effective bandwidth between these

Table 3.3: Video aggregations.

	Videos	Start Times (sec)
	Mr Bean	300
	Aladdin	600
Video Aggregate 1	Jurassic park	900
	Dusk Till Dawn	1200
Video Aggregate 2	Die Hard III	1500
	The Firm	1800
	The Firm	2100
	Mr Bean	2400
	Aladdin	2700
Video Aggregate 3	Jurassic Park	3000

two values, large measurement intervals can result in an unsuccessful search as the measured peak throughput of the trace may be too low.

We investigated the correlation between the setting of the error region and the performance of the binary search algorithm with regards to the number of iterations required to find a suitable effective bandwidth value. We demonstrated that the smaller this region was set, the more iterations of the algorithm were required to find an appropriate solution. The predicted number of iterations for a particular search space has been shown to follow that of the theoretical binary search algorithm of $\log_2 M$. Where M is the number of intervals the search space between 0 and 1 is divided into. We suggest that an error region with β set to 0.01 be used for the purpose of the proposed experiments; this is to ensure that an appropriately accurate effective bandwidth value is estimated within a reasonable amount of algorithm iterations.

An important setting for the algorithm with regards the effective bandwidth estimations collected, is the duration of the packet trace being analysed or T_{max}. We demonstrated through experimentation, that T_{max} must be set in correspondence with the time-scale of the traffic optimisation strategy employing the effective bandwidth estimations. If T_{max} was set to a relatively long period (such as 1 hour +), the resultant effective bandwidth estimation would not reveal the variations in effective bandwidth throughout that particular hour and therefore would not be suitable for use in admission control strategies dealing with service requests arriving at shorter time intervals. On the other hand if T_{max} was set to a shorter duration (seconds), the detailed variation in effective bandwidth may not be necessary to aid long term QoS network planning.

The proposed empirical algorithm was evaluated against two traffic model based approaches that depend heavily on static traffic model assumptions. A variation in traffic model assumptions was simulated through the statistical multiplexing of traffic flows. The completed experiments have demonstrated that as traffic aggregation increases,

Table 3.4: Packet trace details.

Trace	Mean (bps)	Stdev (bps)	Duration (sec)	QoS Target ($delay_{max}$ sec, p_{delay})
MOME-1	163195.51	17544.62	23818	(0.1, 0.01)
MOME-2	216518.49	22306.46	25300	(0.1, 0.01)
MOME-3	25600626.22	3470095.09	899.79	(0.1, 0.01)
MOME-4	22430059.97	3060990.21	899.77	(0.1, 0.01)
CAIDA	760399263.30	112056206.7	63.01	(0.1, 0.01)
Video Aggregate 1	1550322	49752.34	3600	(0.01, 0.0001)
Video Aggregate 2	3651875	49499.49	3600	(0.01, 0.0001)
Video Aggregate 3	7309584	64708.76	3600	(0.01, 0.0001)

the effective bandwidth required to support the QoS targets of a traffic flow decreases. This has been shown to be a non-linear relationship and is different for both elastic and non-elastic traffic. For the Gaussian and Roberts estimation methods, there is no change in the relationship between the estimated effective bandwidth and the mean throughput as the aggregation in traffic flows increases. The main reason behind this is the fact that these approaches assume a high level of aggregation within the traffic being measured; therefore, if the traffic does not conform to their assumptions, the effective bandwidth estimations will be inaccurate. However a case can be made for using model based estimation approaches over an empirical one; for example in the situation of highly aggregated traffic, where effective bandwidth estimations are required at a high frequency, these estimates can be calculated quickly and with sufficient accuracy. The main advantage of our proposed approach is that it is aggregation agnostic.

We finally performed an evaluation of the affect variations in the QoS targets themselves have on the estimation of effective bandwidth of aggregated traffic flows. Experiments have shown that the effective bandwidth of traffic can vary depending on the QoS violation targets set. Based on these observations, the estimation of effective bandwidth alone can not completely control the QoS of traffic on the network, setting these QoS targets to suit the applications being controlled is just as important. If the QoS target of violation is set too loose, an under provisioning of bandwidth may adversely affect the performance of the application, even if QoS targets are being maintained.

The results attained from this chapter have demonstrated the effective bandwidth estimation algorithm based on empirical analysis can be used in various QoS related scenarios for dependable QoS control of aggregated traffic of varying degrees. The advantage of this approach is that QoS control based on these estimations ensures an appropriate amount of bandwidth is reserved to maintain QoS targets within a communications network.

Table 3.5: Effective bandwidth estimations and mean / effective bandwidth ratios.

Trace	Empirical EB (bps)	Empirical EB/Mean Ratio	Gaussian EB (bps)	Gaussian EB/Mean Ratio
MOME-1	19637133.01	120.32	210833.18	1.29
MOME-2	13131186.61	60.64	277085.63	1.28
MOME-3	68297526.29	2.66	35022725.72	1.36
MOME-4	66877733.94	2.98	30741346.41	1.37
CAIDA	1723906364.30	2.26	1064657417.62	1.40
Video Aggregate 1	5912292	3.81	1752923	1.13
Video Aggregate 2	9284482	2.54	3853447	1.05
Video Aggregate 3	10893675	1.49	7533533	1.03

Chapter 4

QoSPlan: Establishing input for QoS-Aware Network Planning

The process of network planning typically involves the use of dedicated network monitoring hardware to gather and collate large amounts of network traffic data, which is then analysed to identify an optimal network configuration design reflecting estimated demand and specified QoS requirements. Use of dedicated hardware means that this approach can be relatively expensive, incurring costs in hardware procurement and maintenance, in addition to significant training and operational costs. In this chapter, we propose an alternative process for supplying the network planning process with network demand information, required to support QoS related network planning activities. The process called QoSPlan is designed to address network operator requirements of supplying viable QoS related data to the network planning process at minimal cost. We provide an investigation into the accuracy and cost efficiency of QoSPlan under different traffic scenarios through a comparison with a direct network monitoring system.

This chapter is organised as follows: §4.1 defines the QoSPlan process. §4.2 offers an evaluation of the QoSPlan process, offering advice on various settings to improve performance under typical network and traffic conditions. An economic analysis of a QoSPlan deployment versus a generic direct monitoring system is presented in §4.3. Finally, §4.4 concludes the chapter with a summary of the QoSPlan process and a discussion of the evaluation results attained.

4.1 QoSPlan Process

The objective of QoSPlan is to supply input to a network planning process for use in QoS related network-planning activities at minimal operational and capital expenditure to the network operator. The network planning process generally requires three sources of input (Awduche *et al.*, 1999) to operate:

1. Attributes associated with the current traffic demands on the network that collectively specify their behavioural characteristics;
2. Attributes associated with constraints on resources and services such as QoS targets on traffic and bandwidth requirements to support them;
3. An optimisation framework that plans traffic and resource configurations subject to 1 and 2.

QoSPlan has been developed to establish inputs 1 and 2. For input 1, a common requirement is the establishment of a network wide matrix of traffic demands per traffic class (termed a demand matrix) (Medina *et al.*, 2002). The demand matrix maps the throughput of traffic between two edge node pairs on a network for a particular traffic class. As the objective of QoSPlan is to supply this input to the network demand matrix with minimal additional hardware deployments, we focus on harnessing the network operators' currently deployed systems to establish this input. Network operators traditionally generate revenue through charging for the usage of bandwidth over their network and therefore generally employ a network accounting system. QoSPlan analyses collected accounting system data to establish suitable input to the network planning process. It is designed as an extension to existing network accounting system deployments (that meet particular requirements, as discussed in section 4.1.3).

As discussed in section 2.2.2, estimation of a demand matrix from accounting system data will be limited in the information provided to the network planning process; it will only address input 1 of a network planning process. Therefore, an enhancement of the demand matrix is required to fulfilling requirement 2 of input to a network planning process. Taking the assumption that QoSPlan will be deployed as an extension to such an accounting system, we propose the architecture depicted in Fig. 4.1. As can be seen, at the point of accounting data mediation, additional planning mediation rules

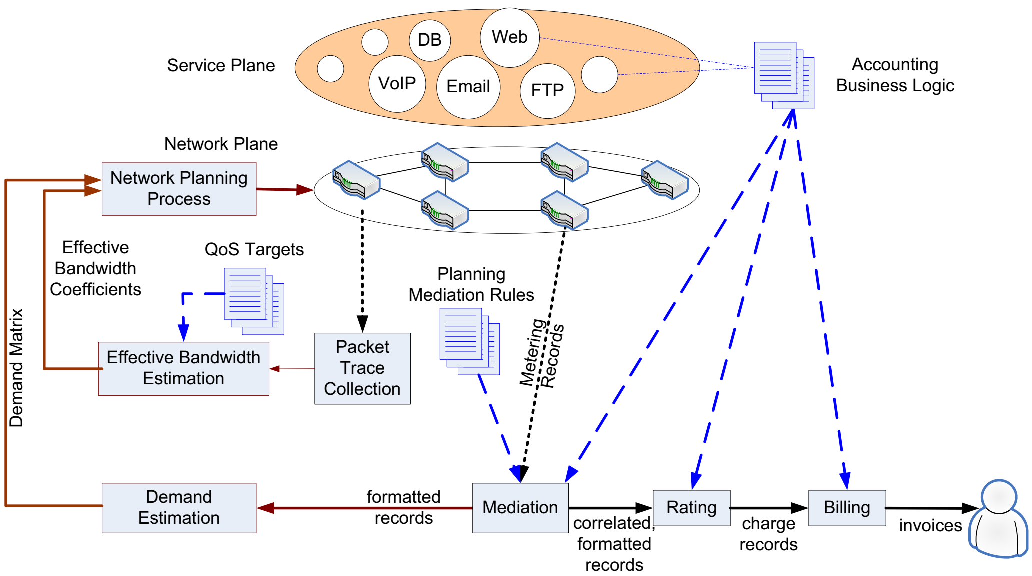

Figure 4.1: QoSPlan Architecture.

are applied to the collected accounting data. This is to ensure accounting data can be formatted into a usable network wide demand matrix.

To support enhancement of the demand matrix, packet traces are collected directly from ingress points of the network. Using supplied QoS targets on packet delay, the packet traces are analysed using the algorithm presented in chapter 3 to provide QoS related effective bandwidth estimations. The following two sections define how QoS-Plan establishes the demand matrix and enhance this information to establish effective bandwidth estimations of traffic associated with supplied QoS targets.

4.1.1 Demand Matrix Estimation Approach

Based on the type of accounting data records available to the network operator, we propose the following mediation process to produce a demand matrix for a network planning process. The process assumes that accounting data records capture traffic demands between source and destination nodes across the network in the form of IP flow records as depicted in Table 4.1; the record in Table 4.1 has a format similar to that of the IETF IPFIX (Sadasivan *et al.*, 2006) and NetFlow 9 (Claise, 2004) accounting records.

The demand matrix estimation algorithm depends on a number of assumptions on

Table 4.1: Sample Accounting Data Flow Record.

SrcAddr	DestAddr	DSCP	Size	Packets	Start	End
10.37.2.22	10.34.1.118	EF	332049	1032	12:35.31	12:48.22

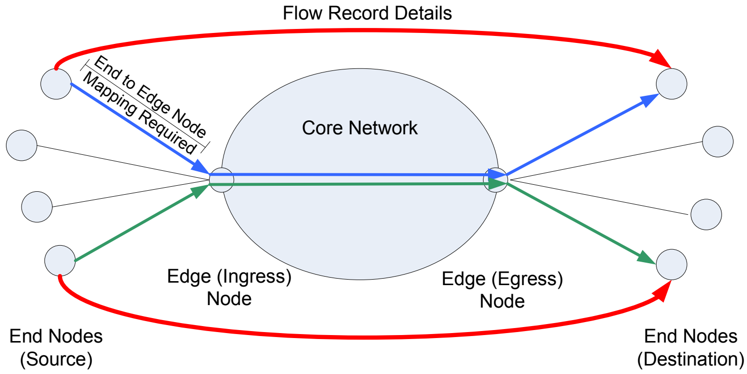

Figure 4.2: Mapping end node flow records to corresponding edge nodes.

accounting records and traffic within the network. It assumes that an end node device[1] is attached to a single edge node[2] of the network. We assume that all traffic entering the network through an ingress edge node, will exit the core network through an egress edge node, therefore no traffic is generated or consumed within the core network. We assume that accounting flow records are only created for traffic entering the network through an ingress edge node. With this, we can be guaranteed that traffic is only recorded once, and that all traffic generated over the network will be accounted for.

The objective of the demand matrix estimation algorithm is to map accounting flow records which only store source and destination edge node information, to their respective edge nodes over which their traffic flowed Fig. 4.2. Once this relationship is identified, the demand of the flow record can be associated with the corresponding edge node.

In estimating network demand from accounting system records, the algorithm makes a number of assumptions. It assumes that packet inter-arrival times and packet sizes within a collected flow record are uniformly distributed. The reason for this assump-

[1]We consider an end node device to be a source or sink of a traffic flow, i.e. the source or destination nodes attached to the network.

[2]We consider edge nodes to be the point of attachment of an end node device to the core network, i.e. the ingress or egress nodes of the network

tion is required is that flow records do not record any information pertaining to the packet size or inter-arrival times of the packets monitored during its creation. Unless clear evidence suggests that another assumption would be appropriate and improve accuracy over the measurement intervals we are interested in, we believe this simplifying assumption will yield appropriate accuracy.

By taking this assumption the proportion of demand within an interval can be easily calculated by multiplying the flows mean rate by the duration of time the flow exists within the current interval. However, this assumption can lead to some inaccuracy in the final estimation process, as throughout the flows duration, packet inter-arrival times and sizes are not normally uniform.

Fig. 4.3 depicts the conditions that our proposed algorithm can estimate network demand from flow records on an interval-by-interval basis. Based on the fact that a flow record contains at the least information such as that in table 4.1, each flow record will have a start time (t_{start}) and an end time (t_{end}). The flow's rate $r(f)$ can be calculated from the flow size divided by the flow duration. As the flow record holds the traffic class the record originated from, each demand matrix can be traffic class specific. The figure shows 4 cases the algorithm captures. The objective of the algorithm is to sum up all demand of all flows that lie within a particular interval $(t, t+t')$.

- Case 1 captures demand of flows that begins before the period and ends during it;

$$d = r(f)(t_{end} - t) \tag{4.1}$$

- Case 2 captures demand of flows that start within the period and ends after it;

$$d = r(f)((t+t') - t_{start}) \tag{4.2}$$

- Case 3 captures demand of flows that start and end within the time period, and finally;

$$d = r(f)(t_{end} - t_{start}) \tag{4.3}$$

- Case 4 captures demand of flows that starts before the period and ends after it.

$$d = r(f)t' \tag{4.4}$$

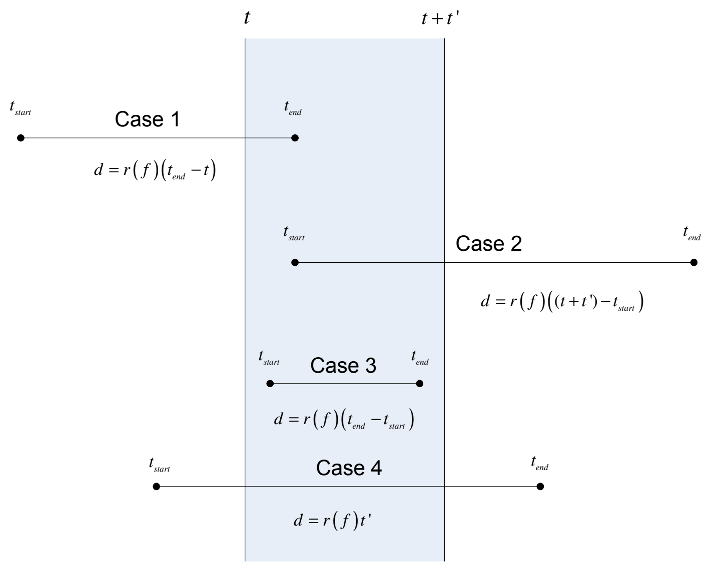

Figure 4.3: Estimation of network demand from collected flow records.

This process defined in Fig. 4.3 is used to estimate demand of a single flow over a particular period. This process is then repeated for all collected accounting records. The objective is to calculate the demand generated by the flow records between two edge node pairs for a particular traffic class. Alg. 4.1 has four nested loops, looping through each ingress router, each metering device, and each end node attached to that metering device, and each flow record within that current metering device. The algorithm matches each flow record to a source node, and estimates the flow's demand within the current measurement interval. The algorithm then matches the destination address of the flow record to a particular egress edge node. This mapping allows us to identify where the traffic is exiting the network.

The findEgressRouter() function in Alg. 4.1 is used to return the egress edge node corresponding to where the flow exits the core network. This function is a simple static table lookup of mappings between the destination end node address within an accounting record and its associated egress edge node. Once the egress edge node is found, the demand estimated for that particular accounting record is added to the appropriate dimension within the demand matrix. Once all records have been processed, the demand matrix is returned. Table 4.2 outlines an example static mapping between edge nodes and end nodes. This can be queried to map an accounting data record to an appropri-

Algorithm 4.1: Demand matrix estimation from accounting flow records.

Input: $EdgeRouters_N$, Interval, TotalTime

Output: DemandMatrix[$IngressRouter, EgressRouter, TOS$]

Init(DemandMatrix); **foreach** *ingressRouter in EdgeRouters* **do**
    foreach *meteringDevice in ingressRouter* **do**
        foreach *sourceNode of meteringDevice* **do**
            foreach *flow of meteringDevice* **do**
                if $flow_{srcip} == sourceNode$ **then**
                    Demand = calculateDemand(timeInterval, interval, flow);
                    egressRouter = findEgressRouter($flow_{destip}$);
                    DemandMatrix[ingressRouter, egressRouter, Interval,
                    $flow_{TOS}$] + = Demand;
return *DemandMatrix*

ate dimension within the demand matrix. Based on this table, for example, the record in Table 4.1 would map to the dimension within the demand matrix for ingress edge node A and egress edge node D. This static lookup table can be created using methods such as interrogation of routing tables within edge routers (Papagiannaki *et al.*, 2004).

4.1.2 Effective Bandwidth Coefficient Estimation Approach

As discussed at the beginning of this section, the demand matrix alone is insufficient for a network planning process to use, as there is no association with the specified QoS targets imposed on traffic by the network operator. We now propose an approach to extend the demand matrix to provide relevant effective bandwidth estimations per traffic class that are specific to QoS targets on packet delay.

The approach is to collect a number of short packet traces from each traffic class at a number of ingress points from around the network. The proposed effective band-

Table 4.2: Mapping between edge nodes and end nodes.

Edge Node	End Point Mappings
A	10.37.1.*, 10.37.2.*, 10.37.3.*
B	10.36.1.*, 10.36.2.*, 10.36.3.*
C	10.35.1.*, 10.35.2.*, 10.35.3.*
D	10.34.1.*, 10.34.2.*, 10.34.3.*

width estimation algorithm as outlined in chapter 3 is used to calculate the effective bandwidth of each packet trace collected. This approach is perfectly suited to this scenario as the effective bandwidth algorithm is independent of any traffic model, and only requires packet traces to operate along with supplied QoS targets on packet delay. A method is devised to relate collected effective bandwidth estimations to the estimated mean throughputs within the demand matrix. The approach taken is to establish a generalised effective bandwidth to mean demand ratio as a method of enhancing the demand matrix. This is termed the effective bandwidth coefficient and is established as follows: the mean throughput, $mean_i$, is calculated for each packet trace collected, as is the associated effective bandwidth for that packet trace $R_{eff,i}$ where i identifies the packet trace being evaluated from the set of collected traces, $i \in \{1..I\}$; using these values we estimate the effective bandwidth coefficient k_i as:

$$k_i = \frac{R_{eff,i}}{mean_i} \tag{4.5}$$

The effective bandwidth coefficient k_i is calculated for all packet traces collected per traffic class. The set of coefficients calculated, allow us to generalise a relationship between estimated network mean demands and effective bandwidth requirements based on supplied QoS targets per traffic class within the network. Further considering a set of I coefficients $k_1, \cdots, k_I$, we first exclude any k_i with too low a mean rate using some appropriate threshold value. The reason for this is that for low levels of traffic aggregation, the effective bandwidth to mean throughput ratio would be quite high in comparison to higher levels of aggregation. This is due to the effect statistical multiplexing has on the effective bandwidth of aggregated traffic flows, as discussed in chapter 3. The contribution of the low mean throughput to overall network demand, is minimal in respect to the effect its associated coefficient may have on the set of coefficients. We then calculate a suitable representative coefficient of effective bandwidth K as the 95^{th} or 99^{th} percentile of the remaining set of coefficients. We believe that K_{95} (the 95^{th} percentile) is an accurate reflection of the mean to effective bandwidth ratio per traffic class and can be used as a method of enhancing the demand matrix.

4.1.3 QoSPlan Process Overview and Considerations

The QoSPlan process is outlined in Fig. 4.4. For a network operator to deploy QoSPlan the following knowledge and capabilities are prerequisites for its adoption:

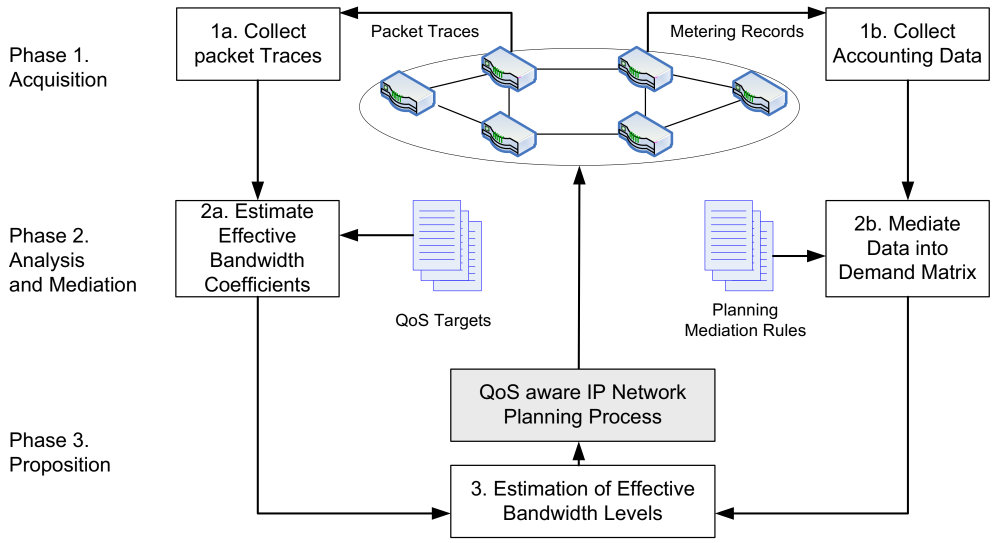

Figure 4.4: QoSPlan process.

1. Deployment of a flow record based accounting system, and access to this data.

2. Information about accounting data collection parameters, such as packet sampling intervals. We demonstrate later that configuration of these parameters can affect the accuracy in estimating network demand from accounting data of different traffic types such as elastic and streaming traffic.

3. The ability to map end nodes to corresponding edge nodes of the network. This is vital in preparing a network wide demand matrix.

4. The ability to collect short packet traces from ingress edge node interfaces for further analysis. This is a requirements for empirical effective bandwidth estimation and the establishment of the effective bandwidth coefficient per traffic class.

5. Knowledge of specified QoS targets being imposed on traffic classes for effective bandwidth estimation.

QoSPlan is broken into three phases: (1) acquisition, (2) analysis and mediation and (3) preparation. We now discuss configuration options at each phase and discuss how important these configurations are in relation to supplying accurate input for network planning.

4.1.3.1 Acquisition

QoSPlan depends on the acquisition of two forms of data, namely accounting data in the form of flow records, and short packet traces. In the collection of flow records, packet sampling plays a major role in the accuracy of demand estimation form accounting data. As discussed in the IETF IPFIX architecture (Sadasivan *et al.*, 2006), PSAMP is employed for packet sampling by IPFIX in the creation of flow records. We analyse the effect different sample settings have on the estimation of network demand from accounting records in section 4.2.

Packet traces are also acquired for input to the QoSPlan process. They are analysed to establish a relationship between network traffic behavior and specified QoS targets through the calculation of effective bandwidth coefficients as discussed in section 4.1.2. A relatively large number of packet traces must be collected per traffic class from various edge node locations around the network. It is vital that collection points be distributed evenly around the ingress points of the network if accurate effective bandwidth analysis is it to be carried out. If the distribution of collection points is uneven, effective bandwidth coefficient estimations per traffic class may be biased to particular ingress points. Packet trace collection points should be positioned at the ingress edge of the network, as we require this traffic to be unshaped by the network itself. Packet traces also need to be collected over an appropriate duration. If packet trace durations are too small or large, analysis of the traffic may result in misleading effective bandwidth estimations as demonstrated in section 3.3.4.4. We recommend packet trace durations of around 4-5 minutes.

4.1.3.2 Mediation and Analysis

The analysis and mediation phase manages collation of relevant metering data into usable information for QoSPlan. There are two internal steps within this phase, namely the estimation of network demand from mediated accounting data, as outlined in section 4.1.1, and the calculation of effective bandwidth coefficients from collected packet traces as outlined in section4.1.2. Both are considered independent processes, but are required to deliver a QoS enhanced demand matrix to a network planning process.

The mediation of accounting data into the demand matrix depends on planning mediation rules, much like accounting record mediation depends on accounting business logic. The planning mediation rules outline how to map accounting record flows to ingress and egress edge nodes on the network. It is important to recognise factors that can affect this mapping such as moving or mobile nodes, or multiple entry points for a particular node. If mappings are not updated appropriately, the demand matrix may contain incorrect measurements. For preparing input to the network planning process, we assume that the network will remain static for the long term. An additional consideration here is the measurement interval over which network demand is being estimated from accounting data. We demonstrate in section 4.2 the effect varying measurements intervals have on the estimation of network demand.

In the analysis of collected packet traces, the effective bandwidth algorithm processes each collected packet trace. Effective bandwidth of a packet trace is controlled by a number of factors such as the traffic itself, degree of aggregation of traffic and most importantly the QoS target. Loose QoS targets can result in an underestimation of required effective bandwidth. Strict QoS targets can results in an over provisioning of resources, which can results in an under utilisation of network resources. Choosing appropriate QoS targets to suit application and end user requirements is therefore a vital consideration for QoSPlan.

4.1.3.3 Preparation

The final phase prepares a matrix of estimated effective bandwidths per traffic class for input to the network planning process. This is achieved by multiplying the appropriate effective bandwidth coefficient by the estimated network demand between edge node pairs for the particular traffic class. A critical decision here is the choice of an appropriate representative effective bandwidth coefficient from the set of collected coefficients per traffic class. As network planning is predominantly based on provisioning for near peak traffic, we recommend choosing the 95^{th} percentile value of this range to ensure a conservative estimate. We analyse this decision later in section 4.2.3.

4.2 Evaluation of a QoSPlan Deployment

We now describe a deployment scenario to evaluate the effectiveness of QoSPlan in supplying input to a network planning process. The objective is to demonstrate that QoSPlan can provide this information with adequate accuracy for long term planning. We use the term long term planning to denote configuring a network for traffic demand and QoS requirements on time scales of days or weeks. We note that for long term planning the level of accuracy required is minimal as planning on this time scale is limited by human usage patterns, which cannot be accurately predicted.

We evaluate a number of configuration options that can be controlled by QoSPlan to improve accuracy for various traffic conditions, such as elastic and streaming traffic, and monitoring conditions, such as packet sampling. As a basis of comparison, we deploy a direct monitoring system within the network to record actual network demand and effective bandwidth levels per traffic class. The following sections discuss the details of our scenario, simulation topology and traffic settings, and finally a set of experiments to test how configuration settings affect accuracy.

4.2.1 Simulation Settings

We propose to evaluate QoSPlan under the following scenario: a single domain network operator offering DiffServ controlled services to subscribed end users with guaranteed QoS targets on packet delay. The network operator uses a deployed IPFIX network monitoring system to supply accounting records to its accounting system for billing purposes. We have simulated a network topology using the OPNET™modeler[1] OPNET (2008); the network topology consists of four core routers, six edge routers and ten workstations (Fig. 4.5). The topology is designed in such a manner as to allow all service traffic to cross the core network through at least one core router. Five workstations operate as servers, with the other five workstations operating as consumers. Each workstation is connected to the network by a10 Mbps Ethernet link. Customers have access to five services; Web browsing, Email, Database, Video on Demand (VoD) and Voice over IP (VoIP). We use the standard OPNET™application models (OPNET,

[1] Each OPNET simulation model was build using existing OPNET network models. An IPFIX device and packet probe device was developed as an add-on to the OPNET router models. The implementations were validated within a number of test case scenarios, which demonstrated expected results

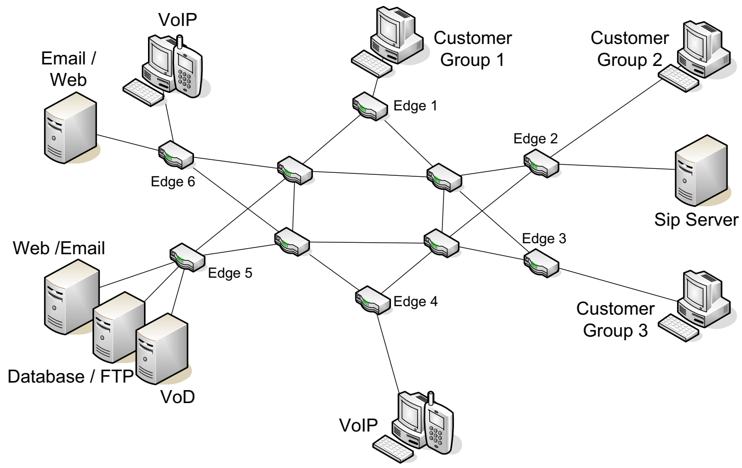

Figure 4.5: QoSPlan simulated network topology.

2008) to model the characteristics of traffic generated by users accessing these services. These traffic models are parameterised to model typical user behaviour in the work place as outlined in Table 4.3. The traffic has been modelled using this approach as we wish to create multiple application interactions across the network between different source and destination pairs; this will ensure an appropriate mix of traffic types and density of flow records collected by the accounting system.

As different applications require different QoS requirements, two DiffServ traffic classes are deployed within the network, one each for elastic and streaming traffic. Services aggregated into Assured Forwarding (AF21) generate elastic traffic and include web and email, and Database. These services specify loose QoS targets on packet delay of (0.04s, 0.001). Voice over IP (VoIP) and Video on Demand (VOD) applications are sources of streaming traffic and are aggregated into Expedited Forwarding (EF) with a QoS packet delay target of (0.02s, 0.0001). Traffic is policed and marked at the ingress routers, where packets are assigned appropriate DiffServ Code Points (DSCPs). Within the core network, all core routers are configured with common Per Hop Behaviour (PHB) settings to ensure all traffic within a particular traffic class is treated the same by each router.

For collecting accounting records, IPFIX monitoring devices are positioned at the ingress interface of all edge routers. The devices deployed are based on the IPFIX archi-

Table 4.3: OPNET traffic model settings.

Traffic Class	Application	Settings
Assured Forwarding	Email	Send/Receive: exp(360 sec)
		Mails sent in groups of : 3
		Email Size : 2000 bytes
		Users per Customer Group : 75
	Web	Page inter-arrival : exp(60 sec)
		Page size : 1500 3000 bytes
		Pages per server : 10
		Users per Customer Group : 75
	Database	Ratio of Queries to other : 100/1
		Transaction inter-arrival : exp(9 sec)
		Transaction size : 1024 bytes
		Users per Customer Group : 125
Expedited Forwarding	VoIP	Codec : G.729 (silence suppression)
		Talk length : exp(0.65 sec)
		Silence length : exp(0.325 sec)
		Users per Customer Group : 10
	VoD	Based on Codec : H.264
		24 frames per sec
		Packet size : 550 - 650 bytes
		Users par Customer Group : 10

tecture (Sadasivan *et al.*, 2006). They can be configured to collect accounting records based on different packet sampling settings. Within the experiments, we evaluate the effect various sampling settings have on the estimation of network demand for elastic and streaming traffic.

For the collection of packet traces, a single monitoring device is modelled as being attached to an ingress interface of an edge router. The device is used to collect packet traces for effective bandwidth analysis. The monitoring device can filter packets from particular DiffServ traffic classes by reading the DSCP within the packet header of each monitored packet. This device can be moved between ingress points around the network, and is only operational during collection.

For experimental comparison to QoSPlan, a direct monitoring system is also modelled. This monitoring system will collect every packet that passes an ingress router interface. These packet traces are used to calculate exact network demand, and effective

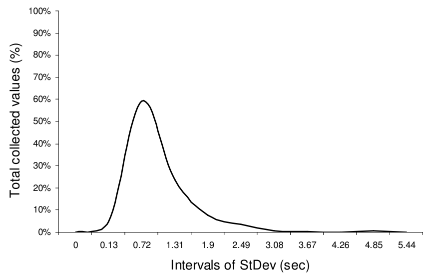

Figure 4.6: Flow duration distribution for AF traffic.

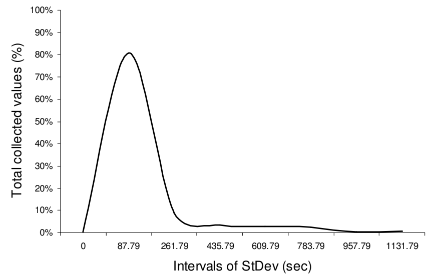

Figure 4.7: Flow duration distribution for EF traffic.

bandwidth estimations within the network.

4.2.2 Captured Flow Record Characteristics

We analyse the characteristics of accounting data flow records collected for both AF and EF traffic classes to demonstrate the difference in characteristics of flow records generated from elastic and streaming traffic. For this comparison, we analyse the distribution of flow durations of the collected records. The traffic was generated between workstations on the network following the traffic model settings in table 4.3. As can be seen in Fig. 4.6 the AF traffic demonstrates a heavy tailed distribution with a mean of approximately 0.8 seconds. On the other hand, the EF traffic has a longer mean duration of approximately 90 seconds with a heavy tail (Fig. 4.7). The heavy tails of these distributions demonstrate the existence of flow records many times the mean, collected for both traffic classes. The knowledge of these flow duration distributions is vital in choosing appropriate demand estimation intervals per traffic class. We use these observations in configuring the QoSPlan demand estimation process later in section 4.2.5

4.2.3 Effective Bandwidth Coefficient Selection

We now study the decision process used in selecting an appropriate representative effective bandwidth coefficient. To achieve this we analyse 1000 effective bandwidth coefficients estimated from collected packet traces for both traffic classes and plot a distribution for each. Figs. 4.8 and 4.9 depict the distribution of effective bandwidth

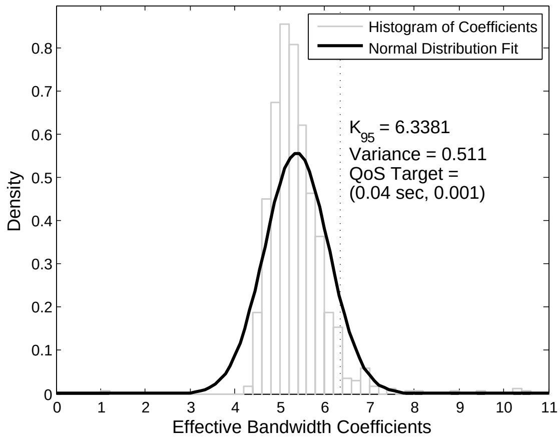

Figure 4.8: Effective bandwidth coefficient distribution of AF traffic.

coefficients collected for AF and EF packet traces respectively. As can be seen, the coefficients approximate a normal distribution.

From this distribution, we must then choose an appropriate value to represent the effective bandwidth coefficient of the associated traffic class. This chosen coefficient will be used to enhance the demand matrix at the final preparation phase of QoSPlan. We recommend choosing the 95^{th} percentile of this range as it represents the relationship between measured mean demand on the network and *near peak* required effective bandwidth for that traffic to maintain outlined QoS targets. Were we to select the mean of the coefficient set, vital mean demand throughput to effective bandwidth relationships will be neglected, leading to underestimation of effective bandwidth levels from the demand matrix. However, in choosing the 95^{th} percentile, we are ensuring that QoSPlan is supplying a network planning process with adequate information to ensure resources are provisioned for the traffic demands on the network. Table 4.4 depicts the chosen effective bandwidth coefficients that will be used to enhance the demand matrix in further experiments.

4.2.4 Deterministic Sampling

This section analyses the effect that deterministic sampling strategies that may be employed by the network accounting system during the collection of accounting data have on the estimation of network demand and estimation of effective bandwidth with

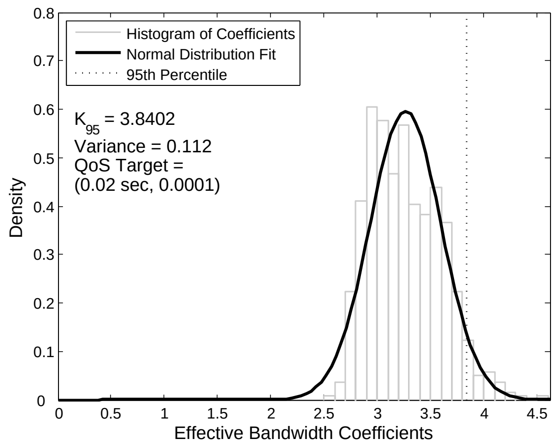

Figure 4.9: Effective bandwidth coefficient distribution of EF traffic.

Table 4.4: Effective bandwidth coefficients used to enhance the traffic class specific demand matrices.

Traffic Class	QoS Delay Target	Coefficient
AF	$(0.04sec, 0.001)$	6.3381
EF	$(0.02sec, 0.0001)$	3.8402

associated coefficients. Packet sampling is assumed to take place on the IPFIX device where packets are monitored and processed into flow records. For QoSPlan to use the sampled accounting data to estimate demand comparable to that of direct measurement approaches, the demand estimated is scaled according to the packet-sampling interval.

In the case of a deterministic packet sampling interval of $n = 1in100packets$ being employed. If an accounting record was calculated to contain a volume of $v = 1Mb$ over its duration, as only 1 in 100 packets would get processed into the flow record, the volume is scaled 100 times to calculate an equivalent volume of $V = 100Mb$, as if all packets within the monitored traffic flow were collected. Scaling of demand in such a manner depends on the assumption of a uniform distribution of packet rate throughout the recorded traffic flow.

A comparison is performed between directly measured network demand between edge router pairs, and demand estimated from accounting data, subject to sampling, collected between the same pair of edge routers over a set interval of 5 minutes. We

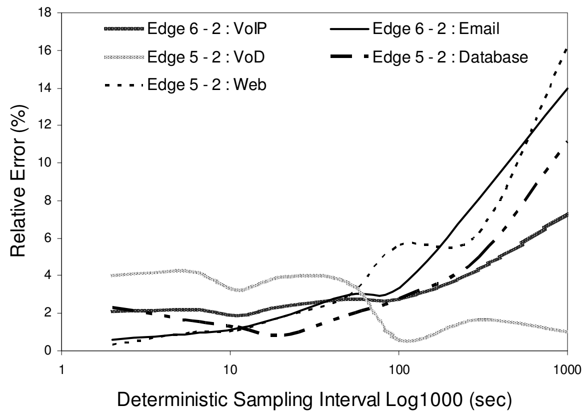

Figure 4.10: Relative error in demand estimation from accounting records collected using deterministic sampling.

perform this experiment for a range of deterministic sampling values and plot the relative error between demand estimated through direct measurement and using the demand matrix estimation algorithm with appropriate scaling (equation 4.6).

$$V = nv \tag{4.6}$$

Fig. 4.10 demonstrates the variation in accuracy of demand estimation from collected accounting data as sampling intervals increase for different application traffic. This demand was calculated based on accounting flow records created using a number of different sampling intervals, where N represents the size of the set a packet is sampled from, i.e. every N^{th} packet is collected. Each flow level demand value calculated was compared to a corresponding direct measurement over the same interval for the same edge node pair. From this a relative error between the two values was estimated and plotted. In this case, the relative error between demand directly measured from the network and estimated network demand from collected flow records increases up to 18% for the AF traffic when a packet-sampling interval $N = 1000$ is used. The main reason behind this degradation in accuracy is due to the scaling of demand from collected accounting data. As elastic traffic tends not to follow a uniform distribution of packet size or inter-arrival times, scaling can skew the results of demand estimation. Therefore, for AF traffic, we recommend a reduced sampling interval of up to $N = 100$

to maintain an acceptable level of accuracy. For EF traffic, relative error manages to remain below 8% even up as far as a packet-sampling interval of 1000. The main reason for this behaviour is down to the fact that streaming traffic tends to have a more uniform distribution of packet size and inter arrival times, lending itself well to the scaling process.

The configuration of this attribute can also affect the accuracy with which QoSPlan estimates effective bandwidth. As the effective bandwidth coefficient QoSPlan uses captures a general relationship between mean demand and effective bandwidth requirements, if mean demand is inaccurately estimated the estimated effective bandwidth will be affected.

We now analyse the relative error between directly measured effective bandwidth values, estimated using packet traces collected every 5 minutes from ingress router interfaces per traffic class, and effective bandwidth estimated with coefficients (in table 4.4) to enhance the demand matrix prepared using different sampling intervals. For each 5 minute interval, the directly measured effective bandwidth is compared with the QoSPlan estimated effective bandwidth and a relative error is estimated.

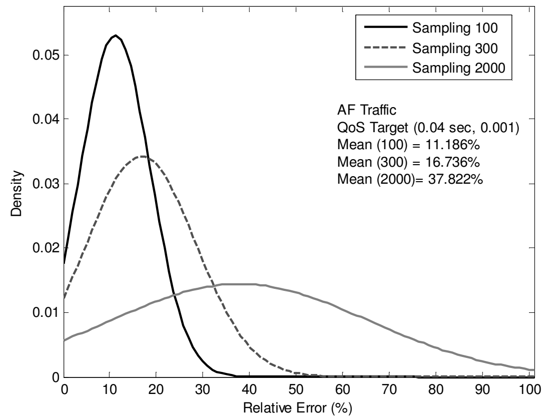

Figure 4.11: Directly measured effective bandwidth versus QoSPlan estimated effective bandwidth for AF traffic.

Fig. 4.11 demonstrates a distribution of global relative error between directly measured and coefficient estimated effective bandwidth values, measured for 1000 different 5-minute intervals. We can see that, for AF traffic, as the packet sampling interval increases, the mean global relative error increases from 11% for N = 100 to 37% for N = 2000. This result demonstrates the effect sampling can have on the estimation of effective bandwidth by QoSPlan using the demand matrix enhanced by effective bandwidth coefficients.

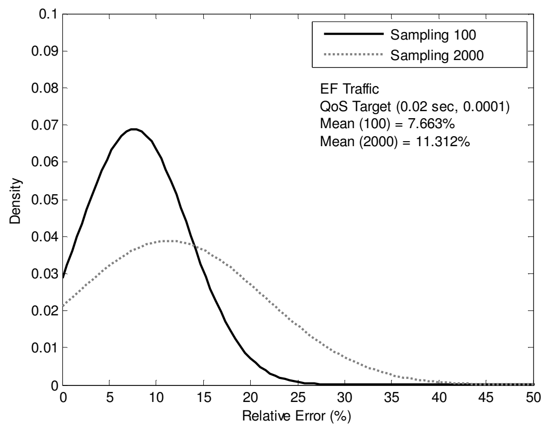

Figure 4.12: Directly measured effective bandwidth versus QoSPlan estimated effective bandwidth for EF traffic.

Fig. 4.12 demonstrates results for EF traffic following the same procedure as above. We can see that for the EF traffic, sampling has a smaller effect on mean global relative error. For a sampling interval of 100, the mean global relative error is approximately 8% while for a much higher sampling interval of 2000, global relative error averages around 11%.

4.2.5 Demand Estimation Interval

The previous experiments have been measuring demand over 5 minute intervals. We now investigate the effect varying this setting has on the demand estimation and demand matrix enhancement accuracy of QoSPlan. We firstly evaluate the estimation of network demand from accounting data versus directly measured demand and graph relative error between the two for different demand estimation intervals. This is performed on both AF and EF traffic. The results from this study are depicted in Fig. 4.13

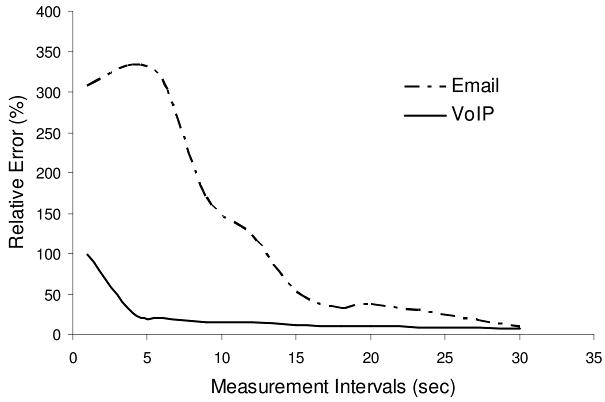

Figure 4.13: Varying the demand estimation interval for AF and EF traffic.

For AF traffic, over very short intervals of below 10 seconds, we see that relative error can be as high as 300% between the average network demand estimated over the interval from accounting data in comparison to direct demand measurements. This high degree of error is attributed to a number of factors including, the type of traffic being monitored, the assumption our demand estimation algorithm makes regarding the packet distribution within flows being uniform, and the process of flow division between measurement intervals (as depicted in Fig. 4.3). As the demand estimation algorithm divides flows proportionally between neighbouring intervals, this can cause a high degree of error in measuring demand per interval for AF traffic, as packets are generally not uniformly distributed within elastic traffic flows. For EF traffic, the case is different as the traffic tends to be more evenly distributed through a flow. This is down to the fact that applications generating EF traffic maintain a relatively steady stream of traffic throughout the duration of the session, such as a video stream, or voice call. Therefore, the division of EF flows into intervals only results in a relative error of less than 50% for measurement intervals of under 10 seconds, with this reducing for larger measurement intervals. We also notice that the relative error reduces by a considerable amount (to below 10%) for measurement intervals greater than 30 second. As within QoSPlan, demand from accounting records will be estimated on the scale of minutes, we maintain that such an assumption of uniform packet distributions within flows is an acceptable assumption. We therefore do not investigate efforts of reducing this relative error.

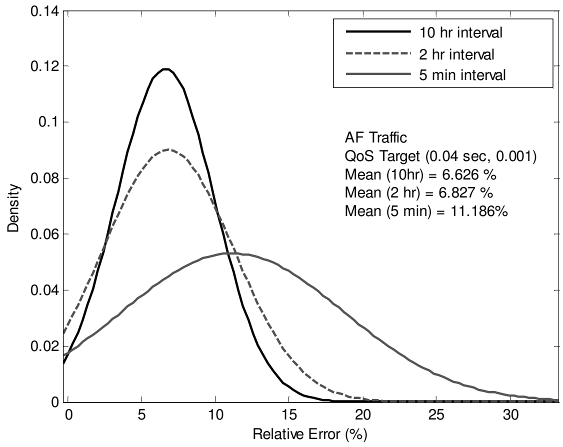

Figure 4.14: Variation in global mean relative error for different demand estimation intervals for AF traffic in comparison to directly measured demand.

We also demonstrate the effect this setting has on the prediction of effective bandwidth levels. We perform a study measuring global relative error between effective bandwidth calculated from the demand matrix enhanced with effective bandwidth coefficients and directly measured effective bandwidth levels. The analysis involves comparing the relative error between the two approaches over different measurement intervals. We plot a distribution of relative error for each set of values recorded. We can use this distribution to estimate a mean relative error between the two approaches, for a specified measurement interval. In Fig. 4.14, we show a mean relative error of around 6% for AF traffic at a measurement interval of 10 hours. The mean remains at just over 6% for a reduced measurement interval of 2 hours. As the measurement interval is reduced to 5 minutes the distribution of relative error increases to a mean of approximately 11%. We see from Fig. 4.15 that for streaming EF traffic there is little variation in relative error over different measurement intervals ranging from approximately 7% for 1 and 2hrs, reaching close to 9.5% for a measurement interval of 10hrs. From this we can see that the longer the measurement interval, the more accurate effective bandwidths can be estimated for both AF and EF traffic.

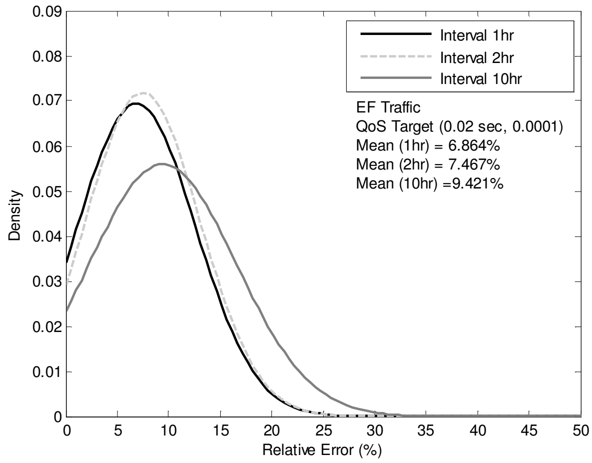

Figure 4.15: Variation in global mean relative error for different demand estimation intervals for EF traffic in comparison to directly measured demand.

4.2.6 Discussion of QoSPlan Scalability

When taking scalability into consideration, there are a number of factors that come into play. We now offer a discussion on these factors and offer arguments in favour of using QoSPlan within ISP networks of various sizes and traffic volumes. These factors include: centralised versus a decentralised deployment; when and where measurements are taken from the network to estimate effective bandwidth coefficients; and available accounting resources for networks of different sizes.

Based on the algorithm specified in Alg. 4.1, a centralised accounting system is assumed. This means that all flow records that are created from the monitored traffic are exported to a single collection module [1]. At this collection module the algorithm will process all records within a particular time interval and estimate demand between identified ingress points and their associated egress points. For small ISPs this assumption is generally valid as multiple collection modules may not be required (this is dependent of the volumes of traffic, and in turn the volume flow records collected). However in a large ISP network, there may be multiple collection modules to which flow records are exported to. In this case the algorithm can be easily distributed so that only

[1] We take the term collection module to mean a centralised location for the storage of accounting records collected from a set of metering devices

records on each of the collectors are processed. One criteria for this process to work is that all collectors have access to the same topological information. This is to ensure all analysed flow records can be associated to an ingress point and an egress point. The resultant demand matrices can simply be added together to prepare a network wide demand matrix. Papagiannaki *et al.* (2004) has already proposed a distributed version of such an algorithm that can easily be ported to our purposes. This ensures the scalability of estimating the demand matrix to large ISP network topologies.

4.3 Economic Analysis of a QoSPlan Deployment

One of the central contentions of this chapter is that a deployment of QoSPlan is significantly more cost effective then a traditional direct monitoring system deployment for supplying input to the network planning process. The previous section demonstrated that QoSPlan can supply QoS related input for long term planning with an acceptable degree of accuracy. We now present the results of a high level comparative economic analysis of the two approaches for three ISPs. Specifically, we compare the costs of extending an existing network accounting system to implement the QoSPlan process with that of a traditional direct monitoring system operating independently of the network accounting system. We proceed by stating our baseline cost assumptions for both deployments. We then state the associated capital and operational costs of both systems, followed by an economic comparison. For a further analysis of cost breakdown with regards ISP expenditure on network management, a recent report (Grabani & Mendel, 2008) has been published discussing the various attributed cost factors. From this report we can see that the ISP spends on average 18% of all management software expenditure on systems supporting network capacity planning activities. The objective of this study is to demonstrate the possible reductions in cost that can be expected with a deployment of QoSPlan by the ISP. We also note that our study is based on incumbent network installations and do not consider the overall costs associated with green field network deployments.

4.3.1 Baseline Cost Assumptions

To form the basis of the economic analysis, we first outline common cost assumptions across the two deployments. For the comparison we assume the network operator has

Table 4.5: Current operational costs per network.

	Small	Medium	Large
Network Nodes	5 edge, 3 core	20 edge, 5 core	45 edge , 15 core
Support Costs	\$20,000	\$60,000	\$200,000
Data Base License Fee	\$20,000	\$60,000	\$200,000
Software License Fees	\$20,000	\$60,000	\$200,000

a deployed network accounting system for usage based accounting. For the currently deployed network accounting system the network operator is required to pay a number of fees, most significantly database system license fees and accounting system software license fees. The network operator also has to pay for customer support for each of these software systems; in general, these fees are set in line with the software license fees. It is common for network operators to incur hardware related costs, for example rental of hardware storage space, hardware-specific support costs, and costs associated with the network operator's replication policy. A replication policy may state that for every one database live within the network, there must be another two database servers replicating every transaction, for redundancy. We assume for this economic analysis that the network operator does not pay rental on hardware space, and does not employ a replication policy. Of course, license fees are typically kept confidential, so the values we choose are based on anecdote. We assume that licence fees increase for larger sized network topologies, which we believe is universally true. Note that our cost model is relatively simplistic; for example we disregard costs such as loss in revenue based on depreciation, down time, data migration.

Given the above assumptions, table 4.5 outlines indicative costs for three ISPs relating to the costs associated with management of their network accounting system. We base the network sizes on existing topologies, including that of the Irish national research and education network HEANET (HEANET, 2008) as a small ISP, and the European wide research and education network GÉANT2 as a comparatively large ISP (GÉANT2, 2008).

Key to our approach is a significant upgrade of the network accounting system to support a QoSPlan deployment. For any significant system upgrade all software, hardware and support costs are likely to increase. We assume that these costs collectively increase by 20%. In addition to the upgrade, specialised contractors must be hired for tasks such as installation, staff training, and on-site support. We assume a contractor charges a flat rate of $1,500 per day, and that he/she can work at a rate of one unit installation per day. However, as the network size increases, so too does the complexity of the installation; thus, we assume that for our medium and large sized networks, contractor fees raise to $1,750 and $2,000 per day, respectively.

Finally, to support new functionality without degrading system performance, additional servers will typically be purchased and deployed; we assume that purchase and deployment of a single server is $5,000. As well as upgrading the accounting system, a QoSPlan deployment also requires use of a limited number of network monitoring devices for collection of traffic traces used in the effective bandwidth estimation process. Based on the geographical size of the network and the duration of time over which network planning is performed, we also must ensure that it is feasible to move the monitoring devices from location to location within an appropriate period of time. For this reason we assume that a maximum of 3 days is reserved for the movement of a device, and that planning is performed over 30 day cycles.

Based on the specified size of the networks and the assumptions on metering device movement restrictions, a small network operator requires 1 such device, for a medium sized network operator we assume 2, and for the large network operators we assume 4 devices are required. We assume that such devices cost approximately $3,000 each. We base this on list prices quoted for such probe devices as the Network Instruments ethernet probe (Network Instruments, 2008). Obviously these prices may vary due to volume purchased, however we simplify this calculation by setting a standard rate for all ISPs. In contrast for the measurement-based network planning approach network monitoring devices must be deployed permanently at all edge routers in the network, resulting in a significant cost overhead for larger network operators.

4.3.2 Comparative Cost Analysis

Based on the cost assumptions outlined above we can now estimate the cost of a QoSPlan and direct monitoring system deployment. We focus on how these deployments affect cost of customer support fees, software license fees, specialised contractor fees, and hardware fees.

Firstly, we address the costs incurred in deploying QoSPlan over an existing accounting system. Additional servers will be required to host the new upgrades to the

Table 4.6: Cost of an incremental QoSPlan deployment.

	Small	Medium	Large
Support Costs	$20,000	$60,000	$200,000
Data Base License Fees	$0	$0	$0
Software License Fees	$4,000	$12,000	$40,000
Server Costs	1 x $5,000	2 x $5,000	4 x $5,000
Network Monitoring Equipment	1 x $3,000	2 x $3,000	4 x $3,000
Contractor Fees	2 x $1,500	4 x $1,750	8 x $2,000
Total Cost	$19,000	$47,500	$128,000

accounting system. As accounting system records are already stored within a deployed database system, there is no need to upgrade the database system. License fees and support costs will be increased by 20% as these are upgraded to an existing system. Depending on network size between 1 and 4 network monitoring devices must be purchased for packet trace collection, and specialised contractors will be required to configure and install them. Table 4.6 shows the cost of an upgrade to QoSPlan for the three network operator types.

Secondly, we address the costs incurred in implementing a traditional direct measurement based approach. The network monitoring system will include the installation of a larger number of network monitoring devices, each monitoring traffic at a single edge router. The deployment will require an extension to the existing database server, as a larger amount of new data will be collected and stored for subsequent analysis; hence database license fees will increase by 20%. Additional servers will be required to host the network monitoring services and applications, which will themselves incur new license and support fees. Finally, installation of the new system and hardware will require specialised contractors. Table 4.7 outlines the cost to the ISP of this approach.

Based on our outlined set of assumptions, table 4.8 shows a clear difference in the cost of deploying both approaches for network of different sizes, with the QoSPlan deployment incurring significantly less costs, particularly as network size increases. This is a result of the greater level of reuse of existing systems in the QoSPlan deployment and the requirement for installation of significantly more hardware in the direct monitoring system deployment.

Table 4.7: Cost of direct network monitoring systems deployment.

	Small	Medium	Large
Support Costs	\$20,000	\$60,000	\$200,000
Data Base License Fees	\$4,000	\$12,000	\$40,000
Software License Fees	\$20,000	\$60,000	\$200,000
Server Costs	1 x \$5,000	2 x \$5,000	4 x \$5,000
Network Monitoring Equipment	5 x \$3,000	22 x \$3,000	49 x \$3,000
Contractor Fees	6 x \$1,500	22 x \$1,750	49 x \$2,000
Total Cost	\$73,000	\$240,500	\$693,000

Table 4.8: Comparison of deployment costs.

	Small	Medium	Large
QoSPlan Deployment	\$19,000	\$47,000	\$128,000
Direct Monitoring Deployment	\$73,000	\$240,500	\$693,000
Difference	\$54,000	\$193,000	\$565,000
(Relative Savings)	74%	80%	82%

4.4 Summary and Conclusions

In this chapter, we have presented QoSPlan, an extension to a traditional usage based network accounting system designed to supply QoS related input to a long-term network planning process. The objective was to achieve this at a reduced cost to the network operator, while maintaining the required level of accuracy. We proposed the QoSPlan process to be followed to ensure the highest level of accuracy is maintained under different traffic conditions, such as elastic and streaming traffic scenarios.

The QoSPlan process depends on two algorithms to supply the required network planning input. Firstly, we presented an algorithm for the mediation of accounting data to supply QoSPlan with a network wide demand matrix. The algorithm depends on the existence of mappings between workstations and their corresponding edge routers. The algorithm also assumes that workstations are static with only one edge router connection to the network. The second algorithm proposed was an approach to enhance the demand matrix with QoS related effective bandwidth requirements specific to outlined QoS targets on traffic. The approach performs an analysis of mean demand to effective bandwidth ratios for a large set of collected packet traces. From this analysis, a representative effective bandwidth coefficient is chosen which provides a method of enhancing the estimated demand matrix.

The QoSPlan process makes a fundamental assumption about collected accounting data records. As these records are collected in the form of IP flow records, the distribution of packet size and arrival times are hidden. QoSPlan therefore assumes a uniform distribution of packets through the collected flow record. Through experimentation, we have shown that because of this assumption key configuration parameters can affect the accuracy of QoSPlan in supplying input.

In the collecting of accounting data, packet-sampling settings can be configured. We have demonstrated that for elastic traffic, as a results of scaling in the demand estimation process, sampling at large intervals of 1000 can result in a relative error of up to 37% in comparison to direct measurement approaches. We have also demonstrated the knock on effect this has on the prediction of effective bandwidth. The reason for this margin of error is a result of the scaling process used to compensate for the sampling of packets. As elastic traffic does not generally follow a uniform distributed throughout a flow, the scaling process can skew results as demonstrated in section 4.2.4. On the other hand we have demonstrated that, for streaming traffic, this margin of error is much lower for sampling intervals of up to 1000. The justification for this is simply down to the fact that streaming traffic behaves more closely to a uniform traffic flow, as assumed by the demand estimation algorithm.

We also demonstrated through experimentation, the effect varying the demand measurement interval has on the estimation of demand from accounting records generated from both elastic and streaming traffic in section 4.2.5. The main source of error, in this circumstance, is the division of demand within flow records between measurement intervals. As can be expected, as demand is not uniformly distributed between flow records generated from elastic traffic, for measurement intervals as low as 10 seconds, we have demonstrated a relative error of up to 300% in comparison to direct measurement results. We have also demonstrated that for elastic traffic, the relative error for the same measurement interval is as low as 50%.

Based on these results, we offer some advice on the configuration of QoSPlan for both elastic and streaming traffic scenarios to ensure the highest level of accuracy achievable. To ensure accurate effective bandwidth predictions as input to a network planning process we recommend using a low sampling interval for the collection of accounting data of around 1 in 100. In the estimation of demand from accounting data, we recommend using a high demand estimation interval of over 1 hour. For streaming traffic, we have shown that configuration settings do not need to be as strict. To reduce processing within the network, a higher sampling interval of 1 in 1000 and higher can be employed in the generation of accounting data. In addition, if a high degree of accuracy is required for estimating demand of streaming traffic, a relatively short demand measurement interval can be used, within the range of minutes (e.g. 5 minutes for an accuracy of within 10%).

Based on an economic analysis, we have demonstrated the clear economic benefits of employing QoSPlan over direct monitoring deployment for the purpose of supplying input to a long term network planning process. As we have demonstrated through experimentation, that QoSPlan can supply this input if properly configured to suit traffic conditions. We have also demonstrated this can be achieved at a fraction of the cost of a full direct monitoring system deployment approach. We have observed that the main advantage of QoSPlan, from an economic perspective, is the reuse of existing infrastructure and collected data.

In conclusion, QoSPlan has been demonstrated to supply adequate input to the network planning process in a cost effective manner, thus achieving the outlined objectives of this chapter. QoSPlan can however, be improved in a number of ways. In the estimation of demand, where QoSPlan currently assumes a uniform packet distribution within recorded accounting data, a method of improving this could be to use more advanced approached to model packet distribution within flows for both elastic and streaming traffic. For long term network planning, we have decided that this step was unnecessary.

Chapter 5

IPTV Admission Control utilising Empirical Effective Bandwidth Estimations

We have previously demonstrated the importance of effective bandwidth estimations for QoS management for long term network planning in chapter 4. We now address the issue of QoS management in the short term and demonstrate how our effective bandwidth estimation algorithm can be leveraged to support QoS control of traffic and improve revenue generation for the service provider.

In the short term, congestion in the network can degrade service performance, which in turn can have a detrimental affect on revenue generated from service usage for the service provider. Admission control is a technique commonly used by service providers to ensure customers' traffic flows are allocated sufficient bandwidth to ensure service level agreement constraints, relating to packet-level Quality-of-Service (QoS), are maintained during periods of high network load. The goal of the service provider is to ensure QoS for accepted traffic flows whilst maximizing bandwidth available for newly arriving flows. For Internet Protocol Television (IPTV) service delivery traffic flows are generally streaming movies or TV programs, which are typically high-bandwidth and have associated with them stringent QoS targets. Therefore, admission control plays a vital role, since bad admission decisions can significantly degrade QoS, not only for the newly accepted flow, but also for already accepted and ongoing flows.

A key feature of any admission control algorithm is how well it predicts the level of resources required to admit a requesting flow. If a new flow request is accepted, the associated packet level QoS targets should be met, without affecting QoS of already admitted flows. Fundamental to achieving this is the accurate estimation of required effective bandwidth of the aggregated traffic already admitted to the network. If the admission control algorithm predicts that effective bandwidth of the aggregate should the flow be accepted will be greater than the available bandwidth, the flow should be rejected. Clearly, the more accurate the estimation of effective bandwidth, the more effectively the admission control algorithm operates. As described in chapter 2, current effective bandwidth estimation approaches are typically based on theoretical analyses of traffic properties (see for example Kelly (1996); Norros *et al.* (1991); Roberts *et al.* (1996)) and make simplifying assumptions such as constant packet sizes and inter-arrival times.

To overcome the limitations of existing effective bandwidth estimation approaches we proposed an empirical approach in chapter 3, which we use in this chapter as the basis of two IPTV-focused admission control algorithms. The first, which we term Empirical Admission Control (EAC), provides an accurate means of predicting the amount of bandwidth required to ensure admitted traffic flows will maintain agreed packet-level QoS targets for the consequent aggregated set of traffic flows. The second, which we term Revenue Maximising Empirical Admission Control (RMEAC), extends EAC by using information relating to the cost, duration and request arrival rate of IPTV content to provide an admission control regime that seeks to maximise the revenue that is generated for the service provider by the accepted flows. We compare the performance of EAC with respect to bandwidth utilisation and QoS control to a number of comparative admission control (AC) algorithms, through simulation of a simple IPTV network focusing on a service provider wishing to control QoS.

This chapter is organised as follows: §5.1 defines the IPTV admission control assumptions we consider. We also define simulation settings used to evaluate the developed admission control algorithms. §5.2 details the EAC algorithm along with our method of evaluation, consisting of the definition of three admission control algorithms for comparison namely Parameter Based Admission Control (PBAC) (Fidler & Sander, 2004; Lima *et al.*, 2004), Experience based Admission Control (EBAC) (Menth, 2004; Menth *et al.*, 2004; Milbrandt *et al.*, 2004, 2006, 2007), and Measurement and Traffic Descriptor based Admission Control (MTAC) (Georgoulas *et al.*, 2004, 2005a). We evaluate the performance of EAC in comparison to PBAC, EBAC and MTAC, showing that the empirical estimation approach facilitates superior admission control decision making. In §5.3 we specify and evaluate the operation of RMEAC, demonstrating its potential to increase revenue in comparison to EAC in situations where there is high demand for more profitable content items. Finally §5.4 summarises the contribution of the chapter and provides concluding remarks.

5.1 IPTV Admission Control Framework

Our admission control framework is founded on estimation of effective bandwidth of traffic by analysis of collected packet traces, as defined in section 3. An effective bandwidth estimation is calculated through analysis of a trace relating to an aggregate of traffic flows collected over a set time interval. This value is used as the estimate of effective bandwidth for that aggregate for the next interval. This estimation provides the information necessary to assess whether admission of flows will result in QoS targets for accepted flows being met. In this section we outline a set of admission control criteria, our IPTV architecture and simulation network and then specify the EAC and RMEAC admission control algorithms.

5.1.1 IPTV Admission Control Requirements

The main decision an admission control algorithm faces is as follows: will the QoS of admitted flows be affected if the requested flow is admitted? The admission control algorithm must look at the current level of bandwidth being consumed by admitted flows to maintain QoS, and predict the required level of bandwidth needed to maintain QoS of all traffic if the flow request is granted admission. This involves predicting future bandwidth requirements of the admitted traffic after admission, with the inclusion of the requesting flow. If the admission control algorithm predicts that the existing traffic flows will not incur QoS violations, the request will be admitted. Therefore, a primary concern of the admission control algorithm is how to predict the level of required resources needed on accepting the flow request.

Before we define our admission control algorithms, we first look at a number of common design strategies. These include the estimation of required bandwidth for the requesting flow, estimation of required bandwidth of admitted flows, and prediction of required bandwidth over the succeeding admission interval.

5.1.1.1 Bandwidth Requirements of Requesting Flows

The most common approach used to estimate the bandwidth requirements of a requesting flow is through the use of a peak throughput rate specified within the flow request. If we take the assumption that the IPTV service provider employs an accounting system based on the IPDR specification as outlined by Cotton & Kleinmann (2006), we can also assume that the service provider abides by the specified accounting and auditing policies, ensuring access to a set of IPTV related content and flow information. This includes peak throughput rates of all available content, cost, duration and flow request frequencies. This information can be inserted into a flow request, and used within the decision process of the admission control algorithm. Other approaches that are used include the measurement of mean throughput rate and standard deviation of throughput for the requesting flow. Such attributes can be included, for example, in the Gaussian admission control algorithm discussed in section 2.2.1.3. The problem here is that such measurements tend not to be available to the service provider.

5.1.1.2 Bandwidth Requirements of Admitted Traffic

The objective of the admission control algorithm is to estimate the bandwidth requirements of the admitted traffic to ensure adequate bandwidth is available to maintain set QoS targets. Without this knowledge, the admission control algorithm will not be able to manage available bandwidth effectively. Approaches that exist include a straight forward peak rate summation of all admitted flows. This method has the advantage of not requiring any additional measurements taken from the network, however this approach does not take into consideration the factor of statistical multiplexing or QoS targets imposed on traffic. Because of this, such an approach tends to be overly conservative in estimating resource requirements.

Other approaches include the use of mathematical models to predict the effective bandwidth of traffic. For example, in the case of a Gaussian approximation (Guérin *et al.*, 1991), the algorithm requires a mean rate and standard deviation measurement of the admitted traffic, along with a QoS related packet loss ratio. These values are then supplied to a mathematical model used to approximate the effective bandwidth of traffic being measured. This approach assumes that the traffic being measured follows a Gaussian distribution. As this only applies to levels of high aggregation, this approach can be quite inaccurate for low aggregates of traffic. For a Roberts approximation (Norros *et al.*, 1991), a mean packet size and inter arrival time of requesting traffic flows, is required for the mathematical model. As the characteristics of traffic flows can vary under a number of different traffic conditions, such as statistical multiplexing of traffic flows, this approach can be inaccurate. These approaches will offer a better estimation of effective bandwidth than a peak rate summation. However, as these approaches rely on static traffic model assumptions, they can quite easily under or over estimate the required effective bandwidth of the traffic being monitored.

We believe the most effective approach is to measure required bandwidth of admitted traffic from measurements alone. If an admission control algorithm were to utilise our proposed effective bandwidth estimation algorithm as detailed in chapter 3, a much more accurate estimation would be achieved. This estimation can be made independent of any traffic model assumptions, will incorporate the affect of statistical multiplexing of traffic has on effective bandwidth estimations, and will be specific to the QoS targets being imposed on the traffic.

5.1.1.3 Prediction of Bandwidth Requirements for Admission Interval

An objective of the admission control algorithm is to predict the required level of bandwidth needed after the time of admission, to accommodate future traffic requirements. A common approach is to use a set of collected bandwidth measurements for the previous n intervals and base the prediction on this range. A number of methods can be used to estimate a future prediction, based on these values such as; a simple averaging over the set, a weighted average applying more significance to newer collected values, and exponentially weighted moving average across the set of values, or simply taking the peak value of the set for a conservative measurement. Further details on these approaches are provided in Appendix. A.

5.1.2 IPTV Simulation Model

From an IPTV admission control perspective, the entities we are interested in are the customer, the service provider and the network operator. A number of projects have proposed business related scenarios for delivery of video-on-demand service over QoS enabled IP networks (EuQoS, 2007; NetQoS, 2007; Vetter *et al.*, 2005). Within these projects, the concept of wholesale bandwidth is used to describe the arrangement where the service provider leases QoS guaranteed bandwidth from the network operator, an arrangement governed via an agreed SLA. In turn, the service provider delivers QoS guaranteed services to its subscribed customers, as governed by a service provider-to-customer SLA (constrained by the limits of its own agreement with the network provider). Failure to meet these SLA targets usually results in loss of revenue through discounting. We define packet level QoS targets related to these agreements within our IPTV scenario and focus on how the service provider ensures its customers receive guaranteed QoS on traffic in line with their SLA(s).

From the above business related scenario, we assume that the service provider has a maximum amount of bandwidth at its disposal. We assume that within this limit, traffic will meet QoS packet delay targets as specified by the service provider to network operator SLA. For our scenario we assume the service provider leases bandwidth from the network operator assuming that at peak times, 20% of its customer base will be accessing offered content items (via unicast video-on-demand). As delivery of content can consume a large quantity of bandwidth, if the assumed peak in access is exceeded without adequate bandwidth management there will be degradation in QoS experienced by all customers, resulting potentially in significant revenue loss and customer dissatisfaction. In times of increasing network load, routers employ techniques such as Random Early Detection (RED), or Weighted Random Early Detection (WRED), which, when throughput reaches a particular threshold within a traffic class, drop packets randomly to avoid congestion. Packets are dropped randomly within the traffic aggregate; therefore, since users are not distinguished between within the traffic aggregate, all users flows will experience degraded QoS. This potential problem is particularly relevant for IPTV, given the relatively strict QoS targets; for example, a packet loss ratio of $10^{-7}s$ is specified by the DSL Forum (Rahrer *et al.*, 2006). The service provider should therefore employ an admission control strategy to avoid QoS violations, whilst maximising bandwidth utilisation to the degree possible.

5.1.2.1 Simulation Topology

We now describe the simulation model used to evaluate the operation of the proposed admission control algorithms. Figure 5.1 depicts a service provider, a network operator

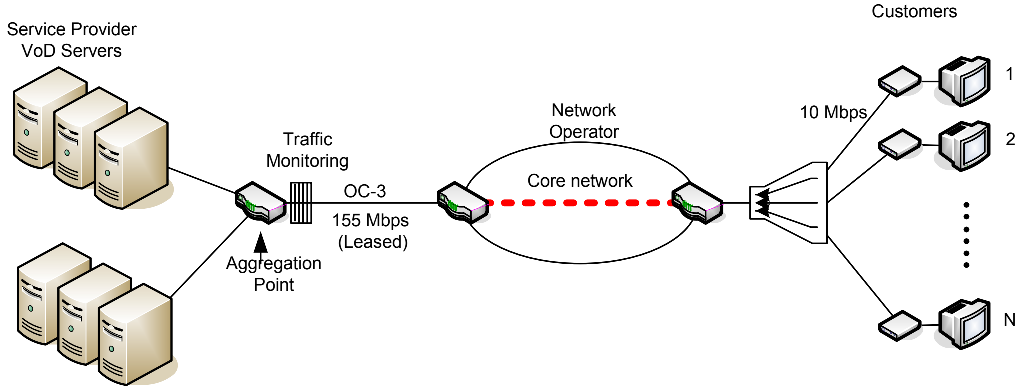

Figure 5.1: IPTV single bottleneck network topology.

and a number of connected customers. Each customer is connected to the network through xDSL with a downstream maximum throughput of 24 Mbps. The service provider has a single point of connection to the network, through which all traffic is aggregated. This is an OC-3 link with maximum bandwidth capacity of $155 Mbps$. It is at this ingress point that the effective bandwidth of aggregated traffic is measured, and the service provider performs admission control on flow requests. All traffic measurement is also performed at this point and we assume that the required network dimensioning and traffic engineering has been performed by the network operator to ensure that traffic within the service providers leased bandwidth capacity will be delivered within agreed SLAs. The service provider wishes to perform admission control on bandwidth up to 90% of this link to ensure there is a 10% margin of precautionary bandwidth available. We assume that the network operator has a DiffServ deployment to manage QoS within the network, and that all video traffic is aggregated into a common traffic class to maintain common QoS requirements on packet delay.

5.1.2.2 QoS Violation Measurement

As part of our evaluation process, QoS traffic violations must be measured per admission control algorithm. We measure QoS violations by collecting a packet trace at a point following admission control and process this trace through a FIFO queue algorithm (the same FIFO Queue algorithm specified in chapter 3). We set the FIFO queue service rate to the maximum reserved bandwidth rate of 139.5Mbps (90% of 155 Mbps). The algorithm will capture the proportion of traffic delayed greater than the specified QoS delay target. If this proportion is greater than the specified violation target, the traffic is deemed in violation of the agreed service level agreement. The algorithm will collect and analyse packet traces of 5 minutes duration, every 5 minutes. We can then monitor the variation in QoS violations over an extended period (hours).

5.1.2.3 User Profiles

The diurnal properties of the user group are modelled using typical assumptions of low and peak periods of activity. Specifically, peak demand periods are assumed to exist during the evening, after work hours, while low demand periods occupy early morning and normal working hours. We assume that all customers are within the same time zone, so that our model represents a reasonably accurate representation of user intensity levels throughout a given 24 hour period. Fig. 5.2 demonstrates our configured content item request arrival rate intensities. Within a particular hour period, we use a standard Poisson arrival process to generate arrivals with the expected mean arrival rate calculated from the corresponding hour period within Fig. 5.2. This model is based on the results of a study of user behaviour in a large scale IPTV network (Yu *et al.*, 2006).

5.1.2.4 Service Popularity and Ranking

The Pareto distribution is commonly used to model a wide range of statistical characteristics such as distribution of wealth within a population. It can also capture other characteristics following the 80 : 20 rule where 80% of traffic generated within the network is contributed to 20% of available services. We follow this rule and model the popularity of our available services on it. The effect this will have on traffic generated is that requests for the most popular content items will contribute to a high percentage of the overall requests.

5.1.2.5 Traffic Models and Characteristics

We use a number of the video frame traces, discussed in Fitzek & Reisslein (2001), to simulate realistic video traffic. The traces have been generated from several video

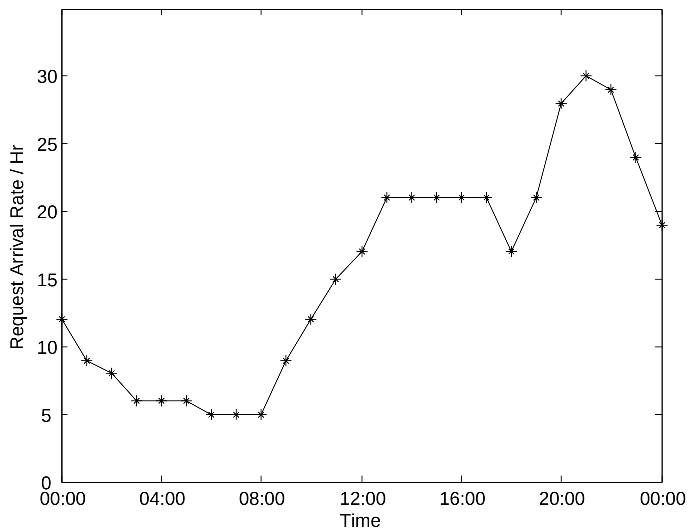

Figure 5.2: Diurnal service arrival rates.

Table 5.1: Service groups and characteristics.

Group	Peak Rate	Cost	Duration
1	2.9Mbps	\$9	$1\frac{1}{2}$ hours
2	2.7Mbps	\$3	45 mins
3	2.6Mbps	\$1	30 mins

sequences of typically 60 minutes duration. For admission control purposes we measure the peak rate of each video trace. Choosing an appropriate interval over which to measure the peak rate of a trace is essential as too large an interval can reduce the accuracy of the estimate, whereas too small an interval can lead to an unnecessarily high peak rate measurements. We choose a measurement interval of $0.05s$ to measure peak throughput, this choice being based on the resolution at which we are measuring Quality of Service violations. Table 5.1. depicts the properties of each service within our simulation. This includes the content item rank, duration, cost and peak rate. An important simplifying assumption we make is that once a request for a content item is accepted, the customer will view that item for its complete duration without breaks (pausing).

5.1.2.6 Pricing Models

A simple pricing model is used where the most popular services generate the most revenue. For example, recently released feature films are typically priced higher then older content items. Within our model, the popularity of a item mirrors its frequency of requests. As requests are based on the Pareto distribution, we use this to provide a pricing structure for the modeled items. Based on this, we divide the items into three groups: group 1 represents the most requested items and are assigned a common cost of $9, the subsequent group 2 have an associated cost of $3, and the remaining group 3 have the lowest cost of $1.

5.2 Empirical Admission Control (EAC) Algorithm

We now specify and evaluate the performance of EAC for ensuring QoS control of admitted traffic while optimally utilising available bandwidth and maximising possible generation of revenue for the service provider respectively. We initially specify the EAC algorithm. We then detail our methodology used to evaluate the EAC, focusing on IPTV scenario considerations and settings, and defining three admission control algorithms EAC will be compared to. We then present a set of experiments aimed to evaluate the performance of EAC within the defined scenario against the other admission control algorithms. The aim of the experiments are to demonstrate the effectiveness of EAC in controlling QoS and utilising available bandwidth effectively at times of high network load.

5.2.1 EAC Algorithm Specification

Based on the requirements stated in section 5.1.1 on admission control algorithm design, we now formally specify EAC. The algorithm receives flow requests for content items which are dealt with on a first come first served basis. Let i^* denote the requesting content item. The peak throughput rate of the requested item is made available to the admission control algorithm, for estimating required bandwidth. Let $p(i^*)$ denote the peak throughput rate of item i^*. An admission control algorithm requires knowledge of the maximum amount of bandwidth leased, to ensure this amount of bandwidth is managed effectively. We denote total bandwidth as B_{TOT}. Let t denote the time of the flow request arrival. The admission control algorithm is required to predict the effective

Table 5.2: Notation used for EAC algorithm definition.

Notation	Description
i^*	The content item for which the flow admission request relates.
$p(i^*)$	The peak throughput rate of item i^*.
B_{TOT}	The total bandwidth leased by the service provider.
$(t, t+t')$	The interval of time for which effective bandwidth is estimated.
$B_{eff}(t, t+t')$	Estimated effective bandwidth for interval $(t, t+t')$.
$\{B_{eff}^N\}$	A set of previously measured effective bandwidth levels.
N	The number of previously measured effective bandwidth levels stored.

bandwidth over the succeeding interval denoted as $(t, t+t')$, where t' is the duration of the interval in question. For prediction of effective bandwidth, the admission control algorithm has access to a set of N previously measured effective bandwidth estimates denoted as $\{B_{eff}^N\}$. These definitions are presented in table 5.2.

To estimate the required bandwidth for the requesting flow, we use the peak rate as supplied by the flow request. This is a common method of predicting bandwidth requirements for requesting flows (Fidler & Sander, 2004; Georgoulas *et al.*, 2004; Milbrandt *et al.*, 2004). We use our proposed effective bandwidth algorithm to estimate the effective bandwidth of admitted traffic. Packet traces are collected from the network, at the point of admission, then analysed by the algorithm. The effective bandwidth estimations calculated by the algorithm are then recorded in a list of the N most recently measured values for EAC to access. EAC then uses these measurements to predict the required effective bandwidth estimation for the following interval. For the prediction of effective bandwidth over the succeeding interval, the approach we take is to use the maximum effective bandwidth value from the recorded set of N effective bandwidth values. We believe this approach ensures we have a conservative prediction of effective bandwidth for admitted traffic. In making this decision we took into consideration the trade off between possible QoS violations incurred if a flow request is accepted under congested conditions versus a complete rejection of the flow request. Generally a rejection response is more acceptable to the customer than an impact on service quality. In addition to this, the possibility of a congested link can affect the service quality of all admitted traffic on the link, potentially affecting the service providers reputation

Algorithm 5.1: Empirical Admission Control (EAC) algorithm

Input: i^*, $\{B^N_{eff}\}$, B_{TOT}

Output: result ($accept|reject$)

Set $B_{eff}(t, t+t') = \max\limits_{\{B^N_{eff}\}} (B_{eff_j} : j = 1, \ldots, N)$;

if $B_{eff}(t, t+t') + p(i^*) \leq B_{TOT}$ **then**

  result $=$ *accept*;

else

  result $=$ *reject*;

return *result*

and thus its business. It is for these reasons that an under-utilisation of bandwidth resources is more preferable to the service provider.

When a flow request is received, the algorithm will evaluate if adequate bandwidth is available on the link to support admission of this flow. If the total link bandwidth minus the predicted estimation of effective bandwidth for the following interval is greater than the bandwidth required by the flow, the request is admitted, otherwise the flow is rejected. We take the assumption that on admission, the flow will not be actively terminated by the service provider or the user.

5.2.2 Admission control algorithms for comparison

In order to evaluate the performance of EAC, we compare its performance in terms of QoS control and bandwidth utilisation against three admission control algorithms. The algorithms attempt to achieve the same objectives as EAC, but following different design principles; they are Parameter Based Admission Control (PBAC) (Fidler & Sander, 2004; Lima *et al.*, 2004); Experience based Admission Control (EBAC) (Menth *et al.*, 2004; Milbrandt *et al.*, 2004, 2007); and Measurement and *a priori* Traffic Descriptor based Admission Control (MTAC) (Georgoulas *et al.*, 2004, 2005a,b). The major difference between their methods and that of EAC is means of estimating effective bandwidth requirements of admitted traffic.

5.2.2.1 Parameter Based Admission Control

As a comparison to the worst case scenario, we model a simple parameter based admission control algorithm (similar in nature to those defined in Fidler & Sander (2004) and Lima *et al.* (2004)). The algorithm's decision depends completely on the peak rate $p(i)$ of content items, as supplied by the traffic descriptor. No measurement is involved in this process. On receiving a content item request, the algorithm decides on admission if the admission test holds in Alg. 5.2. This test is simply whether the sum of the peak rates of all admitted flows, plus that of the new requesting flow, is less than the level of reservable bandwidth, B_{TOT}.

Algorithm 5.2: PBAC admission decision logic

Input: i^*, B_{TOT}, $\{F(t)\}$
Output: result $(accept|reject)$

if $p(i^*) + \sum_{j \in \{F(t)\}} p(j) \leq B_{TOT}$ **then**
  result $= accept$;
else
  result $= reject$;
return *result.*

5.2.2.2 Measurements and *a priori* traffic descriptor based admission control

The Measurement and *a priori* Traffic Descriptor based Admission Control Algorithm (MTAC) estimates the required effective bandwidth of traffic based on the assumption that traffic arriving at the admission point follows Gaussian distribution characteristics. This assumption is based on the work in (Guérin, 1992; Guérin *et al.*, 1991). It uses the Gaussian estimation method, defined is section 3.3.3.1, to estimate the effective bandwidth of aggregated traffic. Initially, the algorithm uses the peak rate of the content item being requested to estimate required bandwidth, however the algorithm also supports using recorded measurements of the mean throughput and standard deviation of the arrival rate of the flow. These values can be incorporated into the estimation of effective bandwidth.

The algorithm also specifies a precaution factor used to ensure the algorithm behaves more conservatively as the network approaches congestion. This factor is calculated as follows; a single reference flow is defined by a reference mean, m_{ref}, and reference standard deviation, σ_{ref}. For the total link capacity, B_{TOT}, the total number of reference flows ,T_{ref}, that can be simultaneously admitted for a given target bound on the Packet Loss Ratio(PLR), is calculated. Using mean and standard deviation measurements collected, the total number of reference flows, N_{ref}, within the admitted traffic is calculated. This is achieved by estimating both the number of reference flows that can produce the measured mean throughput, denoted, N_m, and the number of reference flows that can produce the measured standard deviation, N_σ.

The precaution factor, denoted P_F, defines the relationship between estimated number of reference flows N_{ref} within the measured traffic, and the total number of reference flows T_{ref} allowed within the total link capacity B_{TOT} . If $N_{ref} \leq T_{ref}$ then P_F is set to 1. Admission is simply based on whether the estimated bandwidth multiplied by the precaution factor is less than the total available bandwidth; if not, the flow is rejected (Alg. 5.3).

Algorithm 5.3: MTAC admission decision logic

Input: $p_{new}, B_{TOT}, \epsilon, M_{measured}, \sigma_{measured}, m_{ref}, \sigma_{ref}, T_{ref}$
Output: result $(accept|reject)$
Set $N_{ref} = \dfrac{(N_m + N_\sigma)}{2}$;
Set $N_m = \left\lceil \dfrac{M_{measured}}{m_{ref}} \right\rceil$;
Set $N_\sigma = \left\lceil \dfrac{\sigma^2_{measured}}{\sigma^2_{ref}} \right\rceil$;
if $N_{ref} \leq T_{ref}$ **then**
  Set $P_F = 1$;
else
  Set $P_F = \dfrac{N_{ref}}{T_{ref}}$;
Set $a'_{PLR} = \sqrt{-2ln(\epsilon) - ln(2\pi)}$;
Set $B_{est} = M_{measured} + p_{new} + a'_{PLR}\sqrt{\sigma^2_{measured}}$;
if $(B_{est} X P_F) \leq B_{TOT}$ **then**
  result = *accept*;
else
  result = *reject*;
return *result.*

5.2.2.3 Experience Based Admission Control

The Experience Based Admission Control (EBAC) algorithm operates by taking into consideration peak to mean ratios of currently admitted traffic, denoted $U(t)$. All peak rates of the currently admitted traffic are retained, $F(t)$, along with all the measured mean throughput rate of all existing flows, $M(t)$. Based on a collection of these ratios, the algorithm chooses an appropriate up to date reciprocal of this ration, denoted $\varphi(t)$ (known as the over provisioning factor) suitable for inclusion in its admission control logic. The algorithm gains experience by using a complex approach to estimating the current over provisioning factor based on previous peak to mean ratios. Previous peak to mean ratios are calculated and stored in a time exponentially weighted moving histogram, $P(t, U)$ (Martin & Menth, 2004). By choosing the 95^{th} percentile of this histogram, the most up to date over provisioning factor can be calculated. The approach taken for estimating this value is outlined in Alg. 5.4. The peak measurements used within the algorithm are explicit peak rates, $p(i)$, supplied with the flow request.

To take QoS into consideration, the algorithm sets a maximum link utilisation threshold, ρ_{max}. This threshold is calculated using the Roberts estimation algorithm defined in section 3.3.3.2. The algorithm relies on this probability distribution to estimate the threshold, heavily depending on the queue system assumptions. It uses this to estimate the required maximum link utilisation threshold ρ needed to avoid QoS violations Eq. 5.1. Considering these, the admission control algorithm grants or rejects admission (Algorithm 5.5).

Algorithm 5.4: EBAC computation of the over provisioning factor

Input: $F(t)$, $M(t)$

Output: $\varphi(t)$

Set $R(t) = \sum_{i \in F(t)} p(i)$;

Set $U(t) = \frac{M(t)}{R(t)}$;

Insert $U(t) \rightarrow P(t, U)$;

Set $U_p(t) = \min_u \{u : P(t, U \leq u) \geq p_u\}$;

Set $\varphi(t) = \frac{1}{U_p(t)}$;

return $\varphi(t)$

$$\rho_{max} = \max_{\rho} \rho : P(W > W_{max}) \leq p_W \tag{5.1}$$

Algorithm 5.5: EBAC admission decision logic

Input: $p(i)$, $F(t)$, $\varphi(t)$, ρ_{max}, B_{TOT}
Output: result ($accept|reject$)
Set $R(t) = \sum_{i \in F(t)} p(i)$;
if $p(i_{new}) + R(t) \leq B_{TOT} \cdot \varphi(t) \cdot \rho_{max}$ **then**
  result = *accept*;
else
  result = *reject*;
return *result*

5.2.3 Aggregation of Traffic

A primary advantage of basing admission control on an accurate estimation of effective bandwidth is the ability to account for the statistical multiplexing effect when traffic is aggregated on a link. Awareness of this within EAC allows the algorithm to more accurately predict the level of bandwidth necessary to ensure QoS targets on traffic are maintained, without over estimating bandwidth requirements.

To demonstrate this effect, we set up a simple experiment where 32 video flows are admitted to the network evenly spread over a 5 hour period. We record the prediction of bandwidth made by each algorithm to perform admission control, divided by the current mean throughput. Fig. 5.3 shows the relationship between estimated effective bandwidth coefficient and the number of aggregated flows per algorithm. From this figure we see that:

- PBAC depicts a linear response in the effective bandwidth coefficient as the level of aggregated traffic flows increase. This algorithm predicts required resources based on the sum of peak throughput values per admitted flow. Bandwidth being reserved per traffic flow is on average 6 times that of the mean throughput. This approach does not consider the affect statistical multiplexing has on effective bandwidth requirements. In this case, PBAC results in over provisioning of resources for large aggregates of traffic.

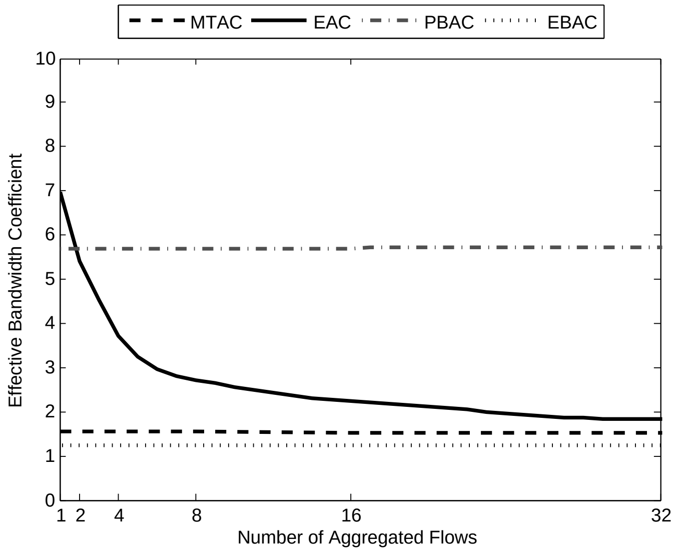

Figure 5.3: Reflection of statistical multiplexing effect in admission control algorithms' bandwidth estimates.

- MTAC demonstrates a strict relationship between measured mean throughput and estimated effective bandwidth of admitted traffic. The ratio is considerably lower than that of PBAC estimates at around 1.3. The algorithm uses a mathematical model to estimate the effective bandwidth of admitted traffic by measuring the mean demand and standard deviation of packet rate of the admitted traffic and supplying these parameters into the model. This approach also demonstrates no obvious reaction to the increase in traffic aggregations and predicted effective bandwidth. This algorithm will under provision bandwidth requirements for low levels of aggregation.

- EBAC demonstrates a similar response to MTAC. As flows are admitted the required bandwidth is estimated based on a selected peak to mean ratio and the maximum utilisation threshold as specified in section 5.2.2.3. This approach also predicts effective bandwidth using a mathematical model and is therefore non responsive to statistical multiplexing of aggregated traffic. As a result, EBAC tends to under provision bandwidth for low levels of aggregated traffic.

- EAC demonstrates a decrease in estimated effective bandwidth as flows are added, observing the affect statistical multiplexing has on aggregated traffic. For a single flow, the ratio of effective bandwidth to mean throughput is around 6. As more traffic is admitted and aggregated together, the ratio reduces. We demonstrate here that EAC can more accurately react to variations in aggregation of traffic, thus provisioning adequate amounts of bandwidth to maintain QoS targets.

5.2.4 Managing trade-off between bandwidth utilisation and QoS violations

Based on the outlined scenario, we perform admission control on the generated flow requests over a number of simulation runs, each using one of the proposed admission control algorithms. The experiments evaluate the proportion of QoS violating traffic and bandwidth utilisation for a set QoS target on packet delay of $(0.02s, 0.0001)$. Fig. 5.4 depicts the bandwidth utilisation based on admission control per algorithm and Fig. 5.5 depicts the QoS violations incurred by the respective algorithms. The results demonstrate the performance of each algorithm in response to the scenario settings outlined in section 5.1.2. We observe that:

- PBAC has the lowest maximum bandwidth utilisation of $\sim 25Mbps$.

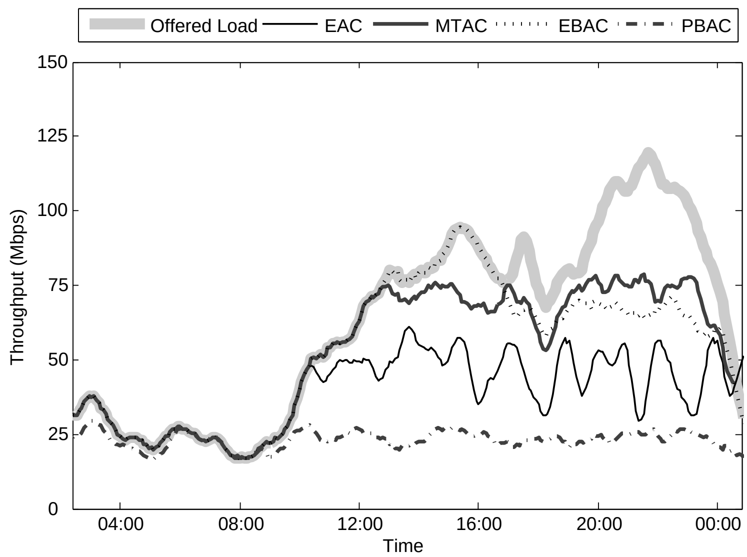

Figure 5.4: Bandwidth utilisation per admission control algorithm.

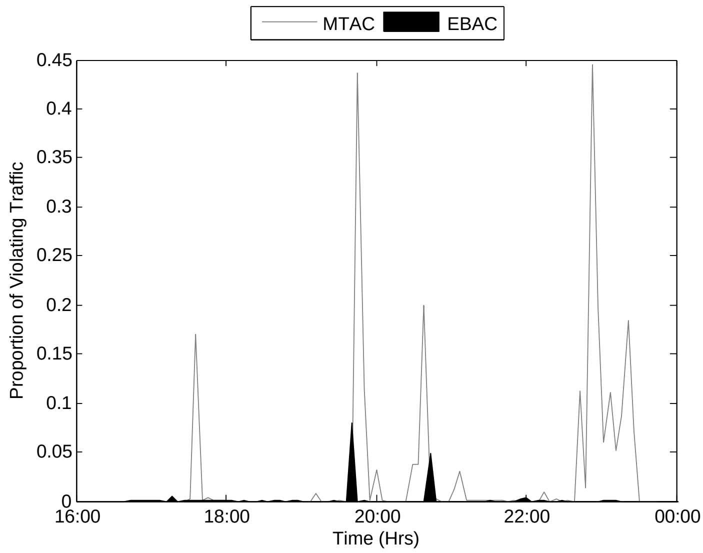

Figure 5.5: QoS Violations per admission control algorithm.

While not incurring any QoS violations, this is considered an undesirable under-utilisation of available resources;

- MTAC has a much higher maximum utilisation of bandwidth at $\sim 75Mbps$. This may be seen as a desirable utilisation of available bandwidth; however this approach does incur significant QoS violations, as can be seen in Fig 5.5. MTAC predicts required bandwidth levels of admitted traffic based on a Gaussian distribution using a packet loss ratio of 0.0001. This leads to an under-estimation of required resources, thus leading to QoS violations as a result of admitting too many flow requests;

- EBAC requires assumptions on traffic in order to estimate required bandwidth levels of aggregated traffic. The algorithm assumes traffic is arriving at a constant packet size and inter-arrival time. Based on analysis of the recorded packet traces we are using we input the following values into the mathematical model: expected packet inter arrival times $E[A] = 0.0131$s, expected packet size $E[B] = 9978.3$ bits. For a QoS target of (0.02s, 0.0001), a maximum link utilisation of 56.3% was calculated. Based on this setting, EBAC maintains a maximum link utilisation of 85Mbps, but also incurs QoS violations;

- EAC utilises an acceptable 50 Mbps of bandwidth (roughly a third of the overall bandwidth available), while incurring no QoS violations for this scenario. These results demonstrate an appropriate control of bandwidth with respect to outlined QoS targets. With respect to the other analysed admission control algorithms, EAC is more conservative than MTAC and EBAC, ensuring admitted traffic is not susceptible to QoS violations. On the other hand, the algorithm is not overly conservative in comparison to PBAC, ensuring a relatively high utilisation of available bandwidth.

5.2.5 Effect of relaxing QoS targets

Within a QoS controlled network, it is important to be able to control the level of violations of set QoS targets. If QoS targets change, the admission control algorithm must respond in kind to the changes. We demonstrate here the effectiveness of EAC in responding to changing QoS requirements. Fig. 5.6 shows for a particular QoS target of $(0.04s, 0.001)$, the utilization of bandwidth for each AC. We observe the following:

- PBAC performs exactly the same as before under the relaxed QoS target as this

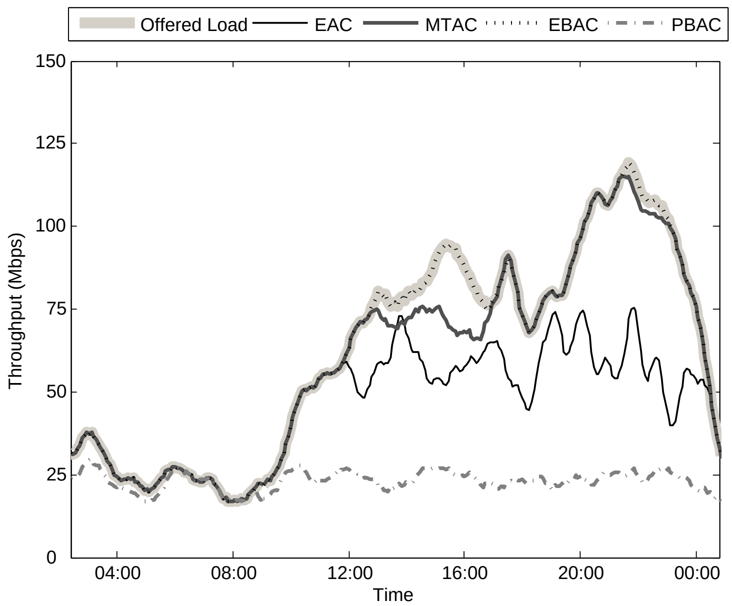

Figure 5.6: Bandwidth utilisation of admission control algorithms with relaxed QoS targets.

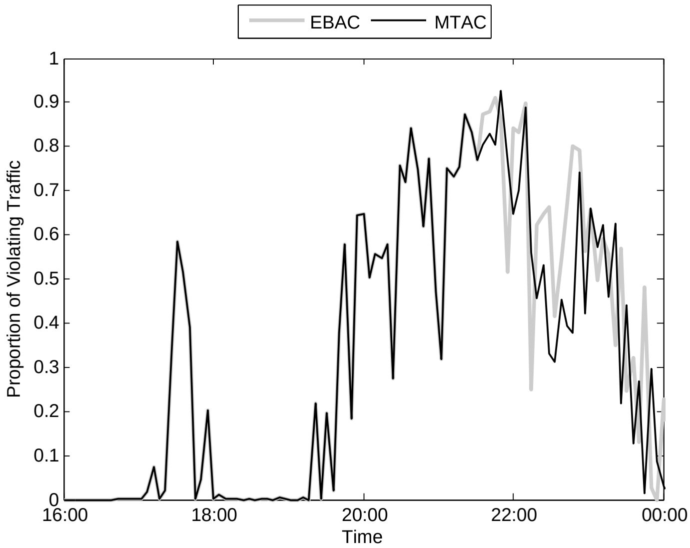

Figure 5.7: QoS violations per admission control algorithm with relaxed QoS targets.

admission control algorithm does not take QoS targets into consideration estimating required bandwidth of admitted traffic. The approach does ensure no QoS violations occur, however demonstrates a poor utilisation of leased bandwidth.

- MTAC uses a relaxed PLR of 0.001 for this experiment. This leads to a bandwidth utilisation of up to 100Mbps, incurring heavy QoS violations as can be seen in figure 5.7 to a peak QoS violation of 0.9 over a 5 min interval. The estimated effective bandwidth for admitted traffic is under estimated using this approach.

- EBAC uses the same packet size and inter arrival time settings as before. The algorithm needs to recalculate the maximum link utilisation threshold for QoS target (0.04s, 0.001). Calculations result in a maximum link utilisation threshold of 80% to control QoS violations for this experiment. The algorithm accepts connections up to this point, which in turn, also incurs heavy QoS violations as depicted in figure 5.7, also up to a peak of 0.9 over a 5 min interval.

- EAC simply requires knowledge of the new QoS target of (0.04, 0.001) to modify its behaviour. The algorithm utilises bandwidth up to 60 Mbps, without incurring any QoS violations. This can be seen as conservative in comparison to the latter two algorithms, but not overly conservative in comparison to PBAC. EAC demonstrates an acceptable balance between QoS control and bandwidth utilisation within this scenario.

5.3 Revenue Maximising Empirical Admission Control (RMEAC) Algorithm

The EAC algorithm is agnostic of both the revenue generated by each flow admission and the revenue lost through flow rejection. We now propose an admission control algorithm that allows knowledge associated with the IPTV content to be used within the admission decision process. The objective is to use historical information about content item request arrivals, together with associated cost and resource requirements, to maximise expected revenue. We base this algorithm on the work of Jennings *et al.* (2001), who defined a revenue optimisation algorithm for load control in Intelligent Network service delivery environments.

5.3.1 RMEAC Algorithm Specification

The algorithm assumes there is a set of I individual items of content made available by the service provider to the customer. Table 5.3 further defines the attributed of the algorithm. Every time a request for item i^* arrives, the admission control algorithm estimates, given its knowledge of the duration and time of acceptance of currently accepted flows, the time interval for which the current level of effective bandwidth of accepted flows (not including that of the new request) will be maintained; this time interval is denoted $(t, t + t')$. Note that the algorithm assumes that no flows are prematurely terminated for any reason.

Each time a request for an item arrives, the admission control algorithm iteratively computes a provisional allocation of the currently unallocated bandwidth to item for interval $(t, t + t')$ in a manner that seeks to maximise the revenue generated for the service provider. Provisional allocations are based on the revenue values for each item, the probability of the arrival of requests for those items in the interval $(t, t+t')$, and the peak bandwidth required for each item. We assume a Poisson process for the arrival of

Table 5.3: Additional notation used for RMEAC

Notation	Description
I	The number of individual items of content available to the Service Provider.
i	A particular content item.
$p(i)$	The peak bandwidth per second required by item i.
$r(i)$	The amount of revenue generated by accepting item i.
T_i	The duration in seconds of flows associated with the streaming of item i.
$q_i(t-t',t)$	The number of requests for item i during the time period $(t-t',t)$.
$n_i(t,t+t')$	The number of requests for item i that have been provisionally allocated by the algorithm.
$u_i(t,t+t')$	The marginal utility of accepting a request for item i during interval $(t,t+t')$.
i^*	The item to which the flow admission request relates.
$v(i)$	The marginal cost associated with provisional allocation of an item i.
$\delta_i(t,t+t')$	The marginal utility per marginal cost of item i.
$\pi_i(t,t+t')$	Variable used to calculate probability of arrivals of an item assuming a poisson arrival process, within each iteration of the algorithm.
$\Pi_i(t,t'_t)$	Variable used to calculate probability of arrivals of an item assuming a poisson arrival process, within each iteration of the algorithm.

requests for items [1], hence the number of arrivals for item i in the interval $(t-t',t)$, denoted $q_i(t-t',t)$, can be taken as an estimate of the number of arrivals for the interval $(t,t+t')$. As the iterations progress, the number of requests for item for which bandwidth has been provisionally allocated, denoted $n_i(t,t+t')$, is stored.

At each iteration the provisional allocation of bandwidth to an item i is the one that maximises the marginal utility to marginal cost in comparison to the other possible allocations. The marginal utility, denoted $u_i(t,t+t')$, is defined as the revenue associated with accepting a request for that item, times the probability of an arrival of an additional request for the item during the interval. If the admission control request is for item i^* then the probability of the arrival of at least one request for item i^* in the interval is set to 1 (since this request has just arrived). The marginal utility can therefore be expressed as:

[1] The Poisson arrival process is 'memoryless' meaning, so the measured mean arrival rate for the previous interval is as good an estimate for the following interval as any.

$$
\begin{aligned}
u_i(t, t+t')|_{i \neq i^*} &= r(i) \sum_{w=n_i(t,t+t')+1}^{\infty} \frac{q_i(t-t',t)^w}{w!} e^{-q_i(t-t',t)} \\
u_{i^*}(t, t+t')|_{n_{i^*}(t,t+t')=0} &= r(i^*) \\
u_{i^*}(t, t+t')|_{n_{i^*}(t,t+t') \neq 0} &= r(i^*) \sum_{w=n_{i^*}(t,t+t')+1}^{\infty} \frac{q_{i^*}(t-t',t)^w}{w!} e^{-q_{i^*}(t-t',t)}
\end{aligned}
\tag{5.2}
$$

The marginal cost associated with provisional allocation of an item i , denoted $v(i)$, is the associated maximum bandwidth consumption of item i over its specified duration:

$$v(i) = p(i)T_i \tag{5.3}$$

The marginal utility per marginal cost of provisionally allocating bandwidth for a request for item i during $(t, t+t')$, is denoted $\delta_i(t, t+t')$, is then:

$$\delta_i(t, t+t) = u_i(t, t+t')/v(i) \tag{5.4}$$

At each iteration the algorithm selects a provisional allocation to an item i', decreasing the currently available bandwidth for the interval, denoted $B(t, t+t')$, by $p(i')$. The algorithm terminates when the provisional allocation is for item $i' = i^*$, in which case the request for item i^* is accepted, or when the value of $B(t, t+t')$ is too small to make a given provisional allocation, in which case the request for item i^* is rejected. The algorithm is formally specified in Alg. 5.6; we first summarise the main stages:

5.3.1.1 Step 1: Initialisation

On the reception of a flow request for item i^*, the algorithm must first calculate t', being the duration until the next admitted flow is schedules to finish. For this period, we assume that the current level of effective bandwidth provisioned for accepted flows will not change. The algorithm will then initilise the appropriate variables.

5.3.1.2 Step 2: Identify Optimal Allocation

In deciding whether to admit request i^*, the algorithm first iterates through the list of available items and calculates the ratio of marginal utility to marginal cost per item. If the requesting item i^* maximises the ratio and there is adequate bandwidth to accept it,

Algorithm 5.6: RMEAC admission control algorithm

Input: $i^*, B_{eff}(t)$

Step 1: Initialization

Calculate $t+t'$ as the time at which the first termination of the currently accepted flows is expected to occur;

forall *items* $i = 1 \dots I$ **do**
    Set $q_i(t-t',t)$ to the number of requests for item i in the interval $(t-t',t)$;
    Set provisional allocation $n_i(t,t+t') = 0$;
    Set $\pi_i(t,t+t') = e^{-q_i(t-t',t)}$;
    Set $\Pi_i(t,t+t') = 1 - \pi_i(t,t+t')$;
    if $i = i^*$ **then**
        Set marginal utility $u_i(t,t+t') = r(i)$;
    else
        Set marginal utility $u_i(t,t+t') = r(i)\Pi_i(t,t+t')$;
    Set marginal cost $v(i) = p(i)T_i$;
    Set remaining bandwidth $B(t,t+t') = B_{TOT} - B_{eff}(t)$;

Step 2: Identify Optimal Provisional Allocation

if $B(t,t+t') > 0$ **then**
    forall *items* $i = 1 \dots I$ **do**
        List all candidates that maximize $\delta_i(t,t+t')$;
    if *list contains item* $i' = i^*$ **then**
        Accept request for item i^*;
        STOP.
    else
        Randomly select candidate allocation i' from the list

Step 3: Perform Allocation

if $B(t,t+t') - p(i') \geq 0$ **then**
    Set $n_{i'}(t,t+t') = n_{i'}(t,t+t') + 1$;
    Set $B(t,t+t') = B(t,t+t') - p(i')$;
else
    Reject request for item i^*;
    STOP.

Step 4: Update Internal Variables

Set $\pi_{i'}(t,t+t') = \pi_{i'}(t,t+t')\dfrac{q_{i'}(t-t',t)}{n_{i'}(t,t+t')}$;
Set $\Pi_{i'}(t,t+t') = \Pi_{i'}(t,t+t') - \pi_{i'}(t,t+t')$;
Set $u_{i'}(t,t+t') = r(i')\Pi_{i'}(t,t+t')$;

Step 5: Loop Statement

if $B(t,t+t') > 0$ **then**
    GOTO Step 2;
else
    Reject request for item i^*;
    STOP.

item is accepted and the algorithm finishes. Otherwise an item i' which does maximise the calculated ratio is notionally allocated the peak bandwidth it would require.

5.3.1.3 Step 3: Perform Allocation

A candidate item i' has been chosen for provisional allocation. The available bandwidth will be updated to reflect the allocation of the peak rate $p(i')$ of the item i'. The counter $n_i(t, t + t')$ is incremented to reflect the provisional allocation of item i'. If there is not enough bandwidth available, item i^* is rejected here, and the algorithm exits.

5.3.1.4 Step 4: Update Internal Variables

If the candidate item i' is accepted, associated variables $\pi_{i'}(t, t+t')$ and $\Pi_{i'}(t, t+t')$ are updated to reflect the probability of another arrival of this item in the time interval t' based on an assumption of exponentially distributed inter arrival times. The marginal utility of this item is also updated. The affect is that, as the number of a particular item increases, the probability of another arrival of this item decreases, thus decreasing its marginal utility.

5.3.1.5 Step 5: Loop Statement

While there is still bandwidth available the algorithm will repeat step 2, otherwise item i^* will be rejected.

5.3.2 Analysis of RMEAC versus EAC

We now wish to evaluate the benefit of deploying RMEAC in comparison to EAC. As RMEAC is an extension to EAC, the estimation of effective bandwidth of admitted traffic is identical. RMEAC is designed to increase the acceptance of high revenue flows over lower revenue flows, while ensuring appropriate bandwidth utilisation with no QoS violations. We perform a set of experiments demonstrating the performance of RMEAC in comparison to EAC with regards to bandwidth utilisation, QoS violations, and revenue gained. The scenario under which we evaluate the algorithms focus on a burst of flow requests with each content item group. We vary the burst intensity for items is each of the three groups to evaluate the performance of RMEAC in dealing with requests of varying value to the service provider. Table **??** outlines the flow settings for this experiment.

5.3.2.1 Burst in flow class 1

We initially simulate a burst in flow requests for content items in group 1. Items within this group have the longest duration at $1\frac{1}{2}$ hours, as well as the most value to the service provider at $9 a viewing. The objective of RMEAC is to ensure that the highest valued items are provisioned for in times of congestion.

Figure 5.8 demonstrates the resulting throughput per content item group over the duration of the experiment. When a burst of requests arrives for item 1, requests for items in the other groups are given less priority based on the RMEAC algorithm. As available bandwidth reduces, all flow requests for items with lower priority are rejected. As a result, during the burst period, 100% of accepted request are for items in group 1. Throughput reaches a maximum utilisation of 50 Mbps, with no QoS violations. When the burst reduces and normal operation is restored, content items from the other lower revenue groups are again granted admission.

To draw a comparison, figure 5.9 has been produced from the exact same experiment conditions with admission controlled by EAC. EAC does not prioritise higher value content items but rather operates on a first come first served basis. The affect here is that a limited number of requests for lower valued content items are admitted, however the loss in revenue is minimal in this case. At the burst period 95% of accepted requests were made up of content item 1, 3% were item 2 and 2% represented item 3. This analysis demonstrates a gain in revenue of 3.7% as shown in table 5.4 if the service provider were to employ RMEAC.

5.3.2.2 Burst in flow class 2

Figure 5.10 depicts the results for a burst in flow requests for content items within group 2. Here the algorithm admits the same level of bandwidth as the previous example. However, instead of traffic for the item group dominating the entire bandwidth, RMEAC preserves flow requests from the higher valued requests for content items within group 1, while rejecting requests for content items from the lower valued group 3. For this scenario, at the burst period 8% of accepted requests were for item 1

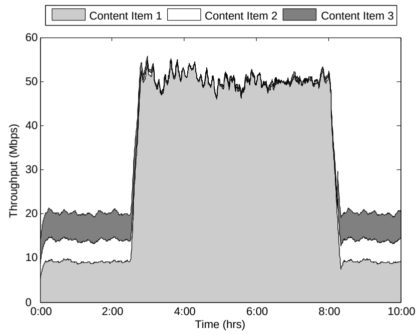

Figure 5.8: Burst in requests for content item 1 (RMEAC)

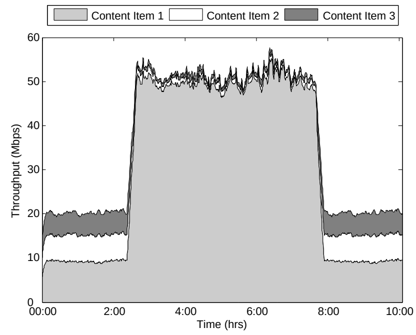

Figure 5.9: Burst in requests for content item 1 (EAC)

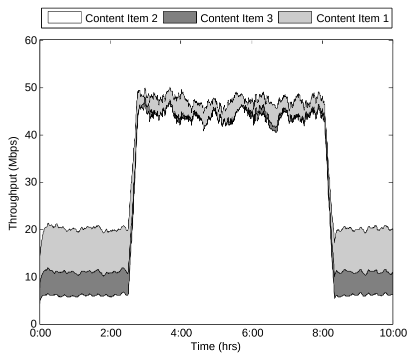

Figure 5.10: Burst in requests for content item 2 (RMEAC)

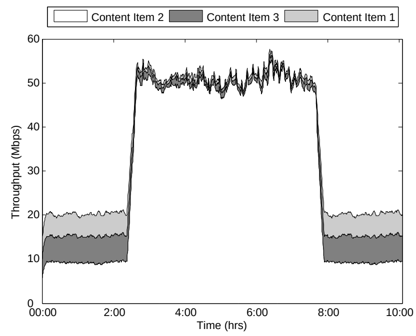

Figure 5.11: Burst in requests for content item 2 (EAC)

and 92% were for item 2 while maintaining up to 50 Mbps bandwidth utilisation and experiencing no QoS violations.

This scenario demonstrates the possible gain in revenue of RMEAC over EAC, as EAC does not priorities higher valued flow requests, and thus loses out on revenue gaining opportunities as depicted on figure 5.11. During the burst period, 2% of accepted requests were for content items within group 1, 95% were for item in group 2 and 3% were for items within group 3. In this case, RMEAC demonstrates a gain of 16.4% in revenue, demonstrating the ability of RMEAC to increase revenue while maintaining QoS control and appropriate bandwidth utilisation.

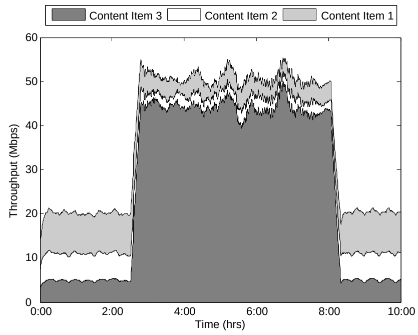

Figure 5.12: Burst in requests for content item 3 (RMEAC)

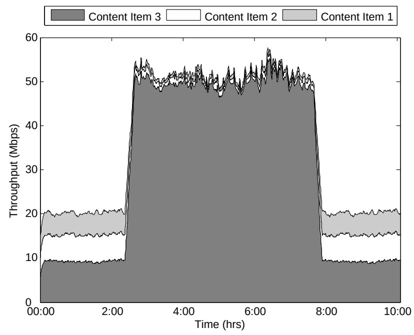

Figure 5.13: Burst in requests for content item 3 (EAC)

5.3.2.3 Burst in flow class 3

Figure 5.12 depicts the bandwidth utilisation of traffic generated by admitted requests from each content item group after a burst in flow requests for content items in group 3. Items within group 3 are the shortest in duration and the lowest in value to the service provider. The results demonstrate that requests from the two higher valued content item groups are maintained as much as possible, thus prioritising these over the lower valued group 3 items. During the burst period, 8% of accepted requests were for content items within group 1, 5% were for items in group 2 and 87% were for items from group 3. As shown in figure 5.13 EAC does not priorities these higher revenue content items, resulting in a reduction in possible revenue gain. Within the burst period 3% of accepted requests were for content items within group 1, 2% were for items in group 2 and 95% were for item in group 3. The contention for RMEAC is simple; if flow requests are to be rejected to maintain bandwidth utilisation and QoS targets, then selectively drop lower value flows to allow higher revenue flow admission. In this case, RMEAC demonstrates a higher gain in revenue of up to 29.7%.

5.4 Summary and Conclusions

In this chapter, we introduced the challenges faced by the IPTV service provider in maintaining QoS control of IPTV traffic in the short term. The service provider's objective is to ensure that the customer receives content within the contractual SLA, even during times of high network load. We focus on how the service provider supports this

Table 5.4: Relative gain in revenue for RMEAC over EAC during times of congestion.

Algorithm		Burst in class 1	Burst in class 2	Burst in class 3
RMEAC	1	100% × 9	8% × 9	8% × 9
	2	0	92% × 3	5% × 3
	3	0	0	87% × 1
EAC	1	95% × 9	2% × 9	2% × 9
	2	3% × 3	95% × 3	3% × 3
	3	2% × 1	3% × 1	95% × 1
Revenue Gain of RMEAC over EAC		3.7%	16.4%	29.7%

objective through admission control. We also bring to light key objectives the service provider wishes to attain in performing admission control on flow requests. The service provider primarily wishes to control QoS of admitted traffic; while utilising bandwidth effectively, and maximise generation of revenue. The main requirement an admission control algorithm needs to fully support these objectives is an accurate estimation of effective bandwidth required to control QoS of admitted traffic. With an accurate estimation of this value, the admission control algorithm can be confident in its admission decisions, while ensuring minimal QoS violations occur with efficient bandwidth utilisation. In line with this contention, we proposed an admission control algorithm based on our empirical effective bandwidth algorithm presented in chapter 3. EAC estimates the effective bandwidth requirements of admitted traffic from packet traces collected from the network. This approach enables the admission control algorithm to account for the effect statistical multiplexing has on the estimated effective bandwidth of aggregated traffic flows. This is vital for an efficient use of leased bandwidth.

We also specified a second admission control algorithm to maximise revenue for the service provider. RMEAC uses additional content related information available to the IPTV service provider to priorities flows that maximise revenue for the network operator at the time of admission. The algorithm uses flow request frequency per content item, cost and duration of the content item in making the admission decision. Using this information the algorithm can protect the admission of higher valued flows in times of high network load while maintaining QoS control and efficient bandwidth utilisation.

Using a set of experiments, we evaluated EAC against three other admission control algorithms under the areas of bandwidth utilisation and QoS violations. The

Algorithm	Traffic Model Assumptions	Aggregation Sensitivity	Bandwidth Utilisation	QoS Violations
EAC	None	Yes	50 Mbps	0
PBAC	None	No	25 Mbps	0
MTAC	Yes	No	75 Mbps	0.44
EBAC	Yes	No	85 Mbps	0.06

Table 5.5: Summary of admission control algorithm results

algorithms chosen represent traditional approaches to admission control and highlight the challenges the IPTV service provider faces with current admission control techniques. Traditional admission control algorithms tend to be designed for a particular problem domain thus depending on tightly defined assumptions on traffic. If traffic deviates from these assumptions, the algorithms begin to under perform and mismanage available bandwidth effectively. The results of the experiments are summarised in table 5.5. The table presents the results for a particular QoS target on video traffic of (0.02 sec, 0.0001). It first states facts about each admission control algorithm with respect to their methods of estimating effective bandwidth of admitted traffic, such as its dependencies on traffic model assumptions and sensitivity to aggregated traffic and the effect on effective bandwidth estimations. The table depicts the results of the analysis, stating maximum bandwidth utilisation and maximum amount of traffic violating the set QoS target over a 5 minute interval. From these results, we see the clear advantage of using EAC over the other algorithms on performing admission control for the service provider. We believe the fundamental advantage of EAC over other approaches is that it ensures an accurate estimation of effective bandwidth to suit the supplied QoS target, and level of aggregation within the admitted traffic, all without the requirement of having a formal traffic model. We do also recognise the drawback of this algorithm in regards to the overhead of collecting and processing packet traces from the network.

Following experimental analysis, we concluded that RMEAC matched EAC with regards to managing bandwidth resources conservatively while ensuring no QoS targets were compromised, as demonstrated in table 5.4. Under the conditions outlined within the experiment, RMEAC enabled a gain in revenue, through protection of higher revenue content thus highlighting its effectiveness in maximising revenue. RMEAC operated better where lower revenue content flooded that network; it ensured that requesting higher revenue content was given priority within the admission decision process, with a gain in revenue demonstrated up to 29.7% under the conditions outlined. This gain is obviously relative to the cost relationship between content items, however it illustrates that revenue can be maximised while ensuring the QoS integrity of traffic and utilisation of bandwidth for the service provider.

Chapter 6

Conclusions and Future Work

The objective of this thesis was to evaluate our hypothesis, which states that a purely empirical approach to the estimation of effective bandwidth of aggregated traffic flows can provide the degree of accuracy required to facilitate effective traffic engineering processes such as QoS-aware network planning and admission control. We now state how the hypothesis was evaluated and state the advantages of the processes and algorithms proposed within this thesis.

In chapter 3 we presented our empirical effective bandwidth estimation algorithm. The algorithm was demonstrated to be capable of accurately estimating the effective bandwidth of aggregated flows of both elastic and streaming traffic. Due to the effect statistical multiplexing has on aggregated traffic flows the algorithm was able to depict a non linear relationship between required effective bandwidth per aggregated traffic flow, and the associated level of aggregation. We demonstrated that traffic model reliant effective bandwidth estimation algorithms evaluated did not capture this relationship, thus demonstrating a clear advantage of using our approach within communications networks.

We evaluated a number of algorithm settings, and highlighted settings suitable for the types of traffic we considered. With regards to an actual deployment of the empirical effective bandwidth estimation algorithm, settings such as an appropriate peak rate measurement interval, error region, and packet trace duration must all be configured to suit the particular traffic scenario. The results and settings demonstrated within this thesis for the algorithms were specific to the traffic traces used within the experiments. This may be considered a disadvantage as a certain amount of off-line configuration is required to ensure suitable results are attained. The approach presented is also only suitable for implementation off-line, this limits its applicability to certain QoS control strategies operating over time-scales of minutes to hours as apposed to millisecond QoS control. The applications we choose to use the proposed algorithm in are inline with this observation, as the required effective bandwidth measurement were over periods of minutes.

In chapter 4 we presented the QoSPlan process. It was developed as a low cost solution for supplying suitably accurate input to a QoS aware long term network planning process. QoSPlan relies on the effective bandwidth algorithm presented in chapter 3 to establish a relationship between required effective bandwidth levels of traffic and an estimated demand matrix. The advantages of QoSPlan are summarised as follows:

- It can accurately estimate the demand matrix of a network from available accounting system data, to within 10 - 15% for the scenarios we considered.
- It can be configured to accurately estimate network demand for both elastic and streaming traffic flows.
- It can provide a demand matrix, enhanced with QoS related characteristics, in contrast to other similar low cost solutions that do not accurately take QoS into consideration.
- It demonstrates a cost saving of up to 80% in operational and capital expenditure comparison to a direct measurement system.

A disadvantage of QoSPlan is that it is dependent on the collected accounting system data. This can have an adverse affect of the completeness of the demand matrix estimated. For example, accounting devices may not be positioned at all ingress routers of the network. In such circumstances, all traffic entering at this edge will not be included within the demand matrix. Another consideration, is if end node workstations have multiple connections to the network (for example, wireless devices). In such a scenario, demand may be mapped incorrectly to corresponding edge routers on the network, therefore leading to an incorrect distribution of demand within the demand matrix. In the estimation of network demand from accounting data, the current algorithm assumes that packets are distributed uniformly through the collected accounting flow record. The estimation of network demand over shorter intervals could be greatly improved, if the algorithm had more knowledge of the packet distribution common to various traffic types, such as elastic and streaming.

In chapter 5 we presented two IPTV focused admission control algorithms based on our effective bandwidth estimation algorithm for accurate QoS control of admitted traffic. The first algorithm, termed EAC, employs a simple evaluation of whether there is sufficient bandwidth available to ensure, with an appropriate degree of confidence, that QoS targets will not be violated if a requested flow is admitted. The second involves a more complex decision process which evaluates the likelihood of the arrival of higher revenue flows in the time it will take to serve the current flow request; if this likelihood is sufficiently high the current request will be rejected even if sufficient bandwidth is available.

EAC was evaluated against three alternative admission control algorithms: PBAC, MTAC and EBAC. Experimental results showed that EAC out-performed these algorithms in terms of reaching the appropriate trade-off between ensuring QoS targets are met and not over-allocating bandwidth. Further results showed that, in certain conditions, RMEAC can out-perform EAC in terms of maximising the revenue generated for the service provider by admitted traffic flows during periods of high network load. We have identified a number of advantages of using the empirical effective bandwidth algorithm within IPTV admission control:

- EAC ensures an accurate estimation of effective bandwidth to suit the supplied QoS target, and level of aggregation within the admitted traffic.
- EAC does not depend on the requirements of a formal traffic model to estimate effective bandwidth.
- As RMEAC uses the empirical effective bandwidth estimation algorithm to control QoS, the service provider can maximise revenue while maintaining QoS targets of admitted traffic.

There are however a number of challenges these algorithms face regarding deployment within an operational communications network. In the process of predicting effective bandwidth from a set of estimated values, the current approach chooses the peak value. This process can be improved to ensure the predicted effective bandwidth values used by the admission control algorithm more accurately represents future demand requirements. The approach used in the reviewed EBAC algorithm, estimated future traffic demands based on a time exponentially distributed moving histogram. Milbrandt *et al.* (2007) demonstrates that such an approach is highly accurate in the presence of traffic change. Such a method could be used to improve the performance of EAC and RMEAC within such scenarios. As the algorithms depend on the collection and analysis of packets traces from the network, they are restricted to a time scale of seconds at best. Such admission control algorithms would not be deemed suitable for admission control strategies within the mili - micro second time frame.

6.1 Future Work

This section proposes a number research areas, that will be considered for future work. The areas relate to: the implementation of the empirical effective bandwidth algorithm itself; a study of the effect various traffic management strategies have on the effective bandwidth of traffic; and an application of the effective bandwidth estimation process within a peer-to-peer scenario.

6.1.1 Applications of Empirical Effective Bandwidth Estimation

As discussed within the thesis, the effective bandwidth estimation is a vital metric for any form of performance optimisation for the control of QoS within the network. We now discuss similar work within applicable fields and state how the proposed approach can contribute to this work.

6.1.1.1 Dynamic Bandwidth Routing

The work proposed by Kandavanam *et al.* (2007) develops an genetic algorithm for dynamic routing of traffic within a communications network. The algorithm operates by dividing the network nodes into clusters, where a Variable Neighbourhood Search (VNS) algorithm is used for route optimisation (Tasgetiren *et al.*, 2004). A genetic algorithm has been employed to optimise the routing of traffic between clustered networks. One advantage of the approach is that it employes a multi-objective optimisation strategy to ensure the route selected adheres to constraints set for a number of different objectives including packet delay, loss, jitter and bandwidth.

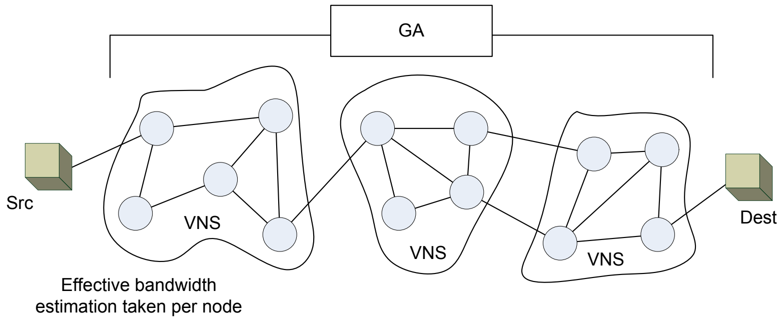

Figure 6.1: Dynamic bandwidth routing using Genetic Algorithms.

The approach relies on the formulation of a demand matrix of the network. Currently, effective bandwidth requirements are not taken into consideration, therefore the optimisation strategies do not take QoS targets into consideration. The research we envisage is based on using effective bandwidth estimation to enhance the demand matrix to ensure the routing optimisation appropriately takes the QoS constraints of traffic into account (Fig. 6.1).

6.1.2 Online Effective Bandwidth Estimation

The current implementation of the purely empirical effective bandwidth estimation algorithm has been developed as an off-line algorithm. This restricts the usefulness of the approach to near-real time use only. The objective of this research would be to develop an online method of empirically estimating effective bandwidth of a traffic flow. The advantage of such an approach would be useful within QoS control strategies such as fast routing, queue management, packet scheduling, and resource optimisation.

Two patents (Lautenschlaeger, 2008; Yip *et al.*, 2005) have been published focusing on this research, each defining an estimation process based on the analysis of a collected packet trace from the network. The approaches each define analysis strategies of packet trace analysis for online estimation. The objective of this research is to develop an online method of effective bandwidth estimation, suitable for use within short term network performance optimisation strategies, such as packet scheduling and queue management. We first intend to evaluate the requirements of such an algorithm, then assess the suitability of our proposed approach in meeting these requirements.

6.1.3 Effective Bandwidth Estimation across Traffic Classes

The work within the thesis focused primarily on measuring the effective bandwidth of traffic aggregated within a particular traffic class. This research would focus on the estimation of effective bandwidth and the effect packet scheduling policies have on the effective bandwidth of aggregated traffic classes. The objective here is to gain a greater understanding of the effect scheduling has on the effective bandwidth of traffic across traffic classes aggregated at a node.

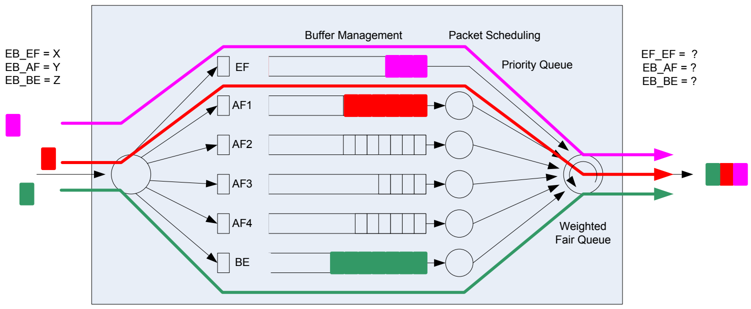

Figure 6.2: Multiple class queueing effect on effective bandwidth.

As depicted in Fig. 6.2, the effective bandwidth per traffic class before the point of aggregation, may be different at the point after aggregation. This would be a result of the dequeueing policies of the node. As packets are scheduled, the characteristics of the traffic within the traffic class can change. A single effective bandwidth value is difficult to measure for the aggregated traffic classes, as they usually require different QoS target constraints. By studying the relationship between packet scheduling and effective bandwidth of traffic classes, current packet scheduling algorithm configurations may be improved, within the objective of guaranteeing QoS of traffic classes within a multi-traffic class network.

6.1.4 Buffer Management for P2P Video Streaming

Within a peer-to-peer live content distribution network, the performance of the distribution network depends, in part, on the optimal configuration of peer node buffers. The issue of P2P buffer design for IPTV live video streaming was highlighted in Hei *et al.* (2006). Peer nodes buffer content based on Groups of Pictures (GoP) within an

MPEG or H.264 video stream. The buffer volume, streaming sequence and scheduling policies all play a part in the overall performance of the peer to peer content distribution network. This content is streamed from one node to the next within set time constraints, as the content being distributed is live (near realtime). As live content can become irrelevant within the context of a live video stream, peer nodes wish to prioritise GoP with respect to timeliness with the video stream. This research intends to take the idea of empirical effective bandwidth estimation and apply this principle to distribution of video content between peer nodes. By setting metric targets on GoPs within a buffer (similar to that of packets within a queue, constrained by a maximum packet delay time) on the distribution of GoP content, buffers can be optimised to ensure these targets are being met for the various types of content being distributed within the network, live video streaming or video on demand or file download.

References

Aboba, B. et al. (2000). Criteria for Evaluating AAA Protocols for Network Access. *IETF Network Working Group RFC 2989*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2989.txt. 28

Almes, G., Kalidindi, S. & Zekauskas, M. (1999a). A One-way Delay Metric for IPPM. *IETF Network Working Group RFC 2679*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2679.txt. 12

Almes, G., Kalidindi, S. & Zekauskas, M. (1999b). A One-way Packet Loss Metric for IPPM. *IETF Network Working Group RFC 2680*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2680.txt. 13

Almes, G., Kalidindi, S. & Zekauskas, M. (1999c). A Round-trip Delay Metric for IPPM. *IETF Network Working Group RFC 2681*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2681.txt. 12

ArborNetworks (2008). PeakFlow SP. Available: 08-10-2008, online at: http://www.arbornetworks.com/products_platform.php. 16

ArcLabs (2008). Waterford Institute of Technology. Available: 09-10-2008, online: http://www.arclabs.ie. 59

ATIS (2007a). A Framework for QoS Metrics and Measurements in Support of IPTV Services. Tech. Rep. ATIS-0800004, The Alliance for Telecommunications Industry Solutions, https://www.atis.org/docstore/products.asp?id=22624. 30

ATIS (2007b). IPTV Architecture Requirements. Tech. Rep. ATIS-0800002, The Alliance for Telecommunications Industry Solutions, available: 08-10-2008, online: https://www.atis.org/docstore/products.asp?id=21213. 30

ATIS (2007c). IPTV Architecture Roadmap. Tech. Rep. ATIS-0800003, The Alliance for Telecommunications Industry Solutions, available: 08-10-2008, online: https://www.atis.org/docstore/products.asp?id=22572. 30

ATIS (2007d). IPTV High Level Architecture. Tech. Rep. ATIS-0800007, The Alliance for Telecommunications Industry Solutions, available: 08-10-2008, online: http://www.itu.int/md/T05-FG.IPTV-C-0610/en. viii, 29, 30

ATIS (2007e). IPTV Packet Loss Issue Report. Tech. Rep. ATIS-0800005, The Alliance for Telecommunications Industry Solutions, available: 08-10-2008, online: https://www.atis.org/docstore/products.asp?id=22659. 30

ATSI (2006). IPTV DRM Interoperability Requirements. Tech. Rep. ATIS-0800001.v002, The Alliance for Telecommunications Industry Solutions, available: 08-10-2008, online: https://www.atis.org/docstore/products.asp?id=21212. 30

Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. & McManus, J. (1999). Requirements for Traffic Engineering Over MPLS. *IETF Network Working Group RFC 2702*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2702.txt. 78

Awduche, D., Chiu, A., Elwalid, A., Widjaja, I. & Xiao, X. (2002). Overview and Principles of Internet Traffic Engineering. *IETF Network Working Group RFC 3272*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3272.txt. viii, 8, 9, 11

Blake, S., Black, D. & Carlson, M. (1998). An Architecture for Differentiated Service. *IETF Network Working Group RFC 2475*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2475.txt. 22

Botvich, D. & Duffield, N. (1995). Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. In *Queueing Systems*, vol. 20, 293–320, Springer. 34, 50, 68

BRADEN, R., CLARK, D. & SHENKER, S. (1994). Integrated Services in the Internet Arcitecture: an Overview. *IETF Network Working Group RFC 1633*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc1633.txt. 19, 21

BRESLAU, L., JAMIN, S. & SHANKER, S. (2000). Comments on the performance of measurement-based admission control algorithms. In *Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 1233–1242. 41

BROWNLEE, N. (1997). Traffic Flow Measurement: Experiences with NetraMet. *IETF Network Working Group RFC 2123*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2123.txt. 14

BROWNLEE, N. (1999). RTFM: Applicability Statement. *IETF Network Working Group RFC 2721*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2721.txt. 14

BROWNLEE, N. (2001). Using NeTraMet for production traffic measurements. In *Proc. of IEEE/IFIP International Symposium on Integrated Network Management (IM)*, 213–226. 14

BROWNLEE, N., MILLS, C. & RUTH, G. (1999). Traffic Flow Measurement: Architecture. *IETF Network Working Group RFC 2722*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2722.txt. 14

CAIDA (2008). CAIDA Anonymized 2008 Internet Traces dataset. Available: 08-10-2008, online at: https://data.caida.org/datasets/passive-2008/. 73, 171

CAIDA (2008). Flowscan. Available: 08-10-2008, online at: http://www.caida.org/tools/utilities/flowscan/. 16

CALHOUN, P., LOUGHNEY, J., GUTTMAN, E., ZORN, G. & ARKKO, J. (2003). Diameter Base Protocol. *IETF Network Working Group RFC 3588*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3588.txt. 28

CALIGARE (2008). Caligare flow inspector. Available: 08-10-2008, online at: http://www.caligare.com/newflow/caligare_flow_inspector.php. 16

Campbell, A., Coulson, G. & Hutchison, D. (1994). A quality of service architecture. In *ACM SIGCOMM Computer Communication Review*, vol. 24, 6 – 27. 8

Carrell, D. & Grant, L. (1997). The TACACS+ Protocol Version 1.78. *IETF Network Working Group Internet Draft*, available: 08-10-2008, online at: http://tools.ietf.org/html/draft-grant-tacacs-02.html. 26

Case, J., Fedor, M., Schofstall, M. & Davin, J. (1990). A Simple Network Management Protocol (SNMP). *IETF Network Working Group RFC 1157*, available: 08-03-2008, online at: http://www.ietf.org/rfc/rfc1157.txt. 13

Cedervall, M. (2008). IPTV Architecture; IPTV functions supported by the IMS subsystem. *European Telecommunications Standards Institute - Telecoms & Internet converged Services & Protocols for Advanced Networks*, http://www.etsi.org/. 30

Claise, B. (2004). Cisco Systems NetFlow Service Export Version 9. *IETF Network Working Group RFC 3954*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3954.txt. viii, 15, 16, 79

ClearSight (2008). ClearSight Networks - ClearSight Analyser. Available: 08-10-2008, online at: http://www.clearsight.com/. 38

Comer, D. (2008). *Computer networks and Internets: with Internet applications*. 5th edition, Indianapolis, Ind. Prentice Hall. 18

Cooper, R.B. (1981). *Introduction to Queueing Theory*. Elsevier North-Holland, Inc., New York, 2nd edn. 163

Cotton, S. & Kleinmann, A. (2006). Service Specification - IP Television (IPTV) ,Resource Data Collection, Analysis & Control (RDC) For IP-Based Services. Tech. Rep. 3.5-A.0.0, http://www.IPDR.org. 32, 110

Cotton, S., Cockrell, B., Walls, P. & Givoly, T., eds. (2004). *IPDR.org Document Map and Overview*, vol. 3.5.0.1, available: 08-10-2008, online: http://www.ipdr.org/public/DocumentMap/DM3.5.0.1.pdf. 28

Davie, B., Charny, A., Bennett, J.C.R., Benson, K., Boudec, J.Y.L., Courtney, W., Davari, S., Firoiu, V. & Stiliadis, D. (2002). An Expidited Forwarding PHB (Per-Hop Behaviour). *IETF Network Working Group RFC 3246*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3246.txt. 23

Davy, A., Botvich, D. & Jennings, B. (2006). An Efficient Process for Estimation of Network Demand for QoS-aware IP Networking Planning. In *Proc. of 6th IEEE International Workshop on IP Operations and Management, IPOM*, vol. LNCS 4268, 120–131, Springer Berlin / Heidelberg. 3

Davy, A., Botvich, D. & Jennings, B. (2007a). Empirical Effective bandwidth Estimation for IPTV Admission Control. In *Proc. of 10th IEEE/IFIP International Conference on Management of Multimedia and Mobile Networks and Services (MMNS)*, vol. LNCS 4787, 125–137, Springer Berlin / Heidelberg. 3

Davy, A., Botvich, D. & Jennings, B. (2007b). On The Use of Accounting Data for QoS-Aware IP Network Planning. In L. Mason & T. Drwiega, eds., *Proc. of 20th International Teletraffic Congress (ITC-20)*, vol. LNCS 4516, 348 – 360, Springer Berlin/Heidelberg. 3

Davy, A., Jennings, B. & Botvich, D. (2007c). QoSPlan: Process for QoS-Aware IP Network Planning Using Accounting Data and Effective Bandwidth Estimation. In *Proc. of 2007 IEEE Global Telecommunications Conference (GLOBECOM)*, 2690–2695. 3

Davy, A., Botvich, D. & Jennings, B. (2008). Revenue Optimized IPTV Admission Control using Empirical Effective Bandwidth Estimation. *IEEE Transactions on Broadcasting*, **vol. 54, issue 3 part 2**, 599 – 611. 3

de Veciana, G., Kesidis, G. & Walrand, J. (1995). Resource management in wide-area ATM networks using effective bandwidths. *IEEE Journal on Selected Areas in Communicaions*, **13**, 1081–1090. 35

Demichelis, C. & Chimento, P. (2002). IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). *IETF Network Working Group RFC 3393*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3393.txt. 13

Divakaran, D.M., Murthy, H.A. & Gonsalves, T.A. (2006). Traffic Modeling and Classification Using Packet Train Length and Packet Train Size. In *Autonomic Principles of IP Operations and Management (IPOM)*, vol. 4268 of *Lecture Notes in Computer Science*, 1–12, Springer Berlin / Heidelberg. 10

Duffield, N. (2004). A Framework for Packet Selection and Reporting. *IETF Packet Sampling WG (PSAMP) - Internet-Draft*, available: 08-10-2008, online at: http://www.ietf.org/internet-drafts/draft-ietf-psamp-framework-10.txt. viii, 17

Duffield, N. & Grossglauser, M. (2001). Trajectory sampling for direct traffic observation. In *IEEE/ACM Transactions on Networking*, vol. 9, 280–292, ACM Press, New York, NY, USA. 38

Duffield, N., Lewis, J.T., O'Connell, N. & Toomey, F. (1995). Entropy of ATM traffic streams: a tool for estimating QoS parameters. In *IEEE Journal on Selected Areas in Communicaions*, vol. 13, 981–990. 35

EuQoS (2007). End-to-End Quality of Service Support over Heterogeneous Networks. *Integrated Project ICT FP6*, available: 08-10-2008, online at: http://www.euqos.eu. 112

Faucheur, F.L. & Lai, W. (2003). Requirements for Suport of Differentiated Services-aware MPLS Traffic Engineering. *IETF Network Working Group RFC 3564*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3564.txt. 25, 26

Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J. & True, F. (2001). Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. In *IEEE/ACM Transactions on Networking*, vol. 9, 265–279. 39

Ferguson, P. & Huston, G. (1998). *Quality of Service: Delivering QoS on the Internet and in corperate networks*. John Wiley and Sons, Inc., New York, NY, USA. 33

Fidler, M. & Sander, V. (2004). A Parameter Based Admission Control Algorithm for Differentiated Services Networks. *Computer Networks*, **44**, 463–479. 108, 117, 118, 119

Fitzek, F.H.P. & Reisslein, M. (2001). MPEG-4 and H. 263 video traces for network performance evaluation. *IEEE Network*, **15**, 40–54. 60, 61, 114

FlukeNetworks (2008). Netflow tracker. Available: 08-10-2008, online at: http://www.crannog-software.com/index.php. 16

GÉANT2 (2008). Available: 08-10-2008, online: http://www.geanet2.net/. 101

Georgoulas, S., Trimintzios, P. & Pavlou, G. (2004). Joint Measurement and Traffic Descriptor Based Admission Control at Real-Time Traffic Aggregation Points. In *Proc. of IEEE International Conference on Communications*, vol. 4, 1841– 1845. 108, 117, 118

Georgoulas, S., Trimintzios, P., Pavlou, G. & Ho, K. (2005a). Heterogeneous Real-time Traffic Admission Control in Differentiated Services Domains. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, 523–528. 108, 118

Georgoulas, S., Trimintzios, P., Pavlou, G. & Ho, K. (2005b). Measurement Based Admission Control for Real Time Traffic in IP Differentiated Services Networks. In *Proc. of 12th IEEE International Conference on Telecommunications (ICT)*, 3–6. 40, 41, 118

Gibbens, R. (1996). Traffic characterisation and effective bandwidths for broadband network traces. In F.P. Kelly, S. Zachary & I.B. Ziedins, eds., *Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Note Series 4*, 169–179, Oxford University Press. 35

GigamonSystems (2008). Gigamon systems - gigavue-mp. Available: 08-10-2008, online at: http://www.gigamon.com/. 38

Gong, W.B., Liu, Y., Misra, V. & Towsley, D. (2005). Self-similarity and long range dependence on the Internet: a second look at the evidence, origins and implications. *Computer Networks*, **48**, 377–399. 10, 35

Grabani, J.P. & Mendel, T. (2008). *Market Overview: The IT Management Software Market In 2008*. Forrester Research, Inc. 100

GUÉRIN, R. (1992). A unified approach to bandwidth allocation and access control in fast packet-switched networks. In *Proc. of 11th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, 1–12, IEEE Computer Society Press. 119

GUÉRIN, R., AHMADI, H. & NAGHSHINEH, M. (1991). Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, **9**, 968–981. 36, 41, 62, 71, 110, 119

GUPTA, V. (1985). What is network planning? *IEEE Communications Magazine*, **23**, 10 – 16. 9

HEANET (2008). Higher Education Authority Network. Available: 08-10-2008, online: http://www.heanet.ie/. 101

HEI, X., LIANG, C., LIANG, J., LIU, Y. & ROSS, K. (2006). A Measurement Study of a Large-Scale P2P IPTV System. *IEEE Transactions on Multimedia*, **9**, 1672 – 1687. 144

HEINANEN, J., BAKER, F., WEISS, W. & WROCLAWSKI, J. (1999). Assured Forwarding PHB Group. *IETF Network Working Group RFC 2597*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2597.txt. 23

ISO (1989). Management Framework. *Part 4 of Information Processing Systems Open Systems Interconection Basic Reference Model*, **7498-4**, available: 08-10-2008, online: http://www.iso.org/iso/iso_catalogue/. 25

ITU-T (2004a). General Overview of Next Generation Networks. *International Telecommunication Union*, available: 08-10-2008, online: http://www.itu.int/md/T05-SG04-050215-TD-WP2-0006/en. 30

ITU-T (2004b). General principles and general reference model for Next Generation Networks. *International Telecommunication Union*, available: 08-10-2008, online: http://www.itu.int/md/T05-SG04-050215-TD-WP2-0005/en. 30

ITU-T (2005). Advanced video coding for generic audiovisual services, Series H: Audiovisual and Multimedia Systems. Tech. Rep. ITU-T Recommendation

H.264, International Telecommunication Union, available: 08-10-2008, online at: http://www.itu.int/rec/T-REC-H.264. 30

JAIN, R. & ROUTHIER, S.A. (1986). Packet Trains-Measurements and a New Model for Computer Network Traffic. *IEEE Journal on Selected Areas in Communicaions*, **4**, 986–995. 10

JAMIN, S., SHENKER, S. & DANZIG, P. (1997). Comparison of Measurement-Based Call Admission Control Algorithms for Controlled-Load Service. In *Proc. of IEEE Conference on Computer Communication (INFOCOM)*, 973 – 980. 41

JENNINGS, B., ARVIDSSON, A. & CURRAN, T. (2001). A Token-based Strategy for Coordinated, Profit-optimal Control of Multiple IN Resources. In *Proc. of the 17th International Teletraffic Congress (ITC 17)*, vol. 1, 245–258. 40, 128

KANDAVANAM, G., BOTVICH, D., BALASUBRAMANIAM, S., SUGANTHAN, P.N. & DONNELLY, W. (2007). A multi-layered solution for supporting ISP traffic demand using genetic algorithm. In *Proc. of 2007 IEEE Congress on Evolutionary Computation (CEC)*, 2032–2039. 142

KELLY, F.P. (1996). Notes on effective bandwidth. In F.P. Kelly, S. Zachary & I. Zeidins, eds., *Stochastic Networks: Theory and Applications*, vol. 4, 141–168, Oxford University Press. 1, 33, 34, 48, 108

KISEL, A. (2008). IPTV Architecture; Dedicated subsystem for IPTV functions. *European Telecommunications Standards Institute - Telecoms & Internet converged Services & Protocols for Advanced Networks*, http://www.etsi.org/. 30

KOODLI, R. & RAVIKANTH, R. (2002). One-way Loss Pattern Sample Metrics. *IETF Network Working Group RFC 3357*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3357.txt. 13

LAUTENSCHLAEGER, W. (2008). Effective bandwidth estimation of a bursty packet stream. *European Patent*, available: 08-10-2008, online at: http://www.freepatentonline.com/EP1876757.html. 143

LEE, C.S. & KNIGHT, D. (2005). Realization of the Next Generation Network. In *IEEE Communications Magazine*, vol. 43, 34–41. 30, 32

Leland, W.E., Taqqu, M.S., Willinger, W. & Wilson, D.V. (1994). On the self-similar nature of Ethernet traffic (extended version). In *IEEE/ACM Transactions on Networking*, vol. 2, 1–15, IEEE Press, Piscataway, NJ, USA. 10, 35, 48, 49

Lima, S.R., Carvalho, P. & Freitas, V. (2004). Distributed Admission Control for QoS and SLS Management. In *Journal of Network and Systems Management*, vol. 12, 397–426, Springer. 108, 118, 119

Liu, B., Figueirdo, D.R., Guo, Y., Kurose, J. & Towsley, D. (2001). A study of networks simulation efficiency: fluid simulation vs. packet-level simulation. In *Proc. of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 2001, Anchorage, AK, USA. 37

Liu, N.X. & Baras, J.S. (2004). Measurement and Simulation Based Effective Bandwidth Estimation. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 4, 2108 – 2112. 37, 50, 51

Maglaris, B., Anasstassiou, D., Sen, P., Karlsson, G. & Robbins, J.D. (1988). Performance models of statistical multiplexing in packet video communications. In *IEEE Transactions on Communications*, vol. 36, 834–844. 68

Mainwaring, K. (2007). Requirements for network transport capabilities to support IPTV services. Tech. Rep. ETSI TS 181 014 v2.0.0, European Telecommunications Standards Institute - Telecoms & Internet converged Services & Protocols for Advanced Networks. 30

ManageEngine (2008). Netflow analyzer. Available: 08-10-2008, online at: http://manageengine.adventnet.com/products/netflow/index.html. 16

Mansfield, G., Roy, T. & Shiratori, N. (2001). Self-Similar and fractal nature of Internet traffic data. In *Proc. of 15th International Conference on Information Networking (ICOIN)*, 227–231. 10, 35

Martin, R. & Menth, M. (2004). Improving the Timeliness of Rate Measurements. In *Proc. of GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, VDE Verlag. 121

MARTTI, J. (2007). Service Layer Requirements to Integrate NGN Services and IPTV. *European Telecommunications Standards Institute - Telecoms & Internet converged Services & Protocols for Advanced Networks*, http://www.etsi.org/. 30

MATHIS, M. & ALLMAN, A. (2001). A Framework for Defining Empirical Bulk Transfer Capacity Metrics. *IETF Network Working Group RFC 3148*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3148.txt. 12

MEDINA, A., FRALEIGH, C., TAFT, N., BATTACHARRYYA, S. & DIOT, C. (2002). A taxonomy of IP traffic matrices. In *SPIE proceedings series*, 200–213. 37, 78

MENDENHALL, W., BEAVER, R.J. & BEAVER, B.M. (2005). *Introduction to Probability and Statistics*. Thomson Brooks/Cole, 12th edn. 165

MENTH, M. (2004). *Efficient Admission Control and Routing for Resilient Communication Networks*. Ph.D. thesis, Inst. für Informatik, Lehrstuhl für Verteilte Systeme. 108

MENTH, M., MILBRANDT, J. & OECHSNER, S. (2004). Experience-based admission control (EBAC). In *Proc. of the 9th International Symposium on Computers and Communications (ISCC)*, vol. 2, 903–910, IEEE Computer Society, Washington, DC, USA. 41, 108, 118

MILBRANDT, J., MENTH, M. & OECHSNER, S. (2004). EBAC-A Simple Admission Control Mechanism. In *Proc. of IEEE International Conference on Network Protocols (ICNP)*. 108, 117, 118

MILBRANDT, J., MENTH, M. & JUNKER, J. (2006). Experience-Based Admission Control with Type-Specific Overbooking. In *Proc. of the 6th IEEE International Workshop on IP Operations and Managemen (IPOM)*, vol. LNCS 4268, 72–83, Springer Berlin / Heidelberg. 41, 108

MILBRANDT, J., MENTH, M. & JUNKER, J. (2007). Experience-Based Admission Control in the Presence of Traffic Changes. In *Journal of Communications*, vol. 2, 10–21, Acadmey Publisher. 40, 41, 108, 118, 142

MILLS, C., HIRSCH, G. & RUTH, G. (1991). Internet Accounting Background. *IETF Network Working Group RFC 1272*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc1272.txt. 26, 27

MOERMAN, K., FISHBURN, J., LASSERRE, M. & GINSBURG, D. (2005). Utah's UTOPIA: An Ethernet-Based MPLS/VPLS Triple Play Deployment. In *IEEE Communications Magazine*, vol. 43, 142–150. 32

MOME (2008). IST MOME - Cluster of European Projects aimed at Monitoring and Measurement. Available: 08-10-2008, online at: http://www.ist-mome.org/database. 58, 73, 171

NETQOS (2007). Policy Based Management of Heterogeneous Networks for Guaranteed QoS. *Specific Target Research Project (STREP) IST FP6*, available: 08-10-2008, online at: http://www.netqos.eu. 112

NETUSAGE (2008). Total network visibility. Available: 08-10-2008, online at: http://www.netusage.net/14/. 16

NETWORK INSTRUMENTS (2008). 10/100/1000 Ethernet Probes. Available: 01-10-08, online: http://www.networkinstruments.com/products/probes/. 102

NETWORK-INSTRUMENTS (2008). Network Instruments ,LLC. Observer. Available: 08-10-2008, online at: http://www.networkinstruments.com/. 38

NICHOLS, K., BLAKE, S., BAKER, F. & BLACK, D. (1998). Definition of the Differentiated Services Field (DS Field) in the IPv6 and IPv6 Headers. *IETF Network Working Group RFC 2474*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2474.txt. 22, 24

NORROS, I., ROBERTS, J.W., SIMONIAN, A. & VIRTAMO, J.T. (1991). The superposition of variable bit rate sources in an atm multiplexer. In *IEEE Journal on Selected Areas in Communications*, vol. 9, 378–387. 108, 111

OBJECTPLANET (2008). Network probe. Available: 08-10-2008, online at: http://www.objectplanet.com/probe/. 38

OPNET (2008). Discrete event smulation model library. *OPNET Modeler^{TM}*, available: 08-10-2008, online at: http://www.opnet.com/products/library/des_model_library.hml. 61, 88

Papagiannaki, K., Taft, N. & Lakhina, A. (2004). A distributed approach to measure IP traffic matrices. In *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement*, 161–174. 39, 83, 100

Paxson, V., Almes, G., Mahdavi, J. & Mathis, M. (1998). Framework for IP Performance Metrics. *IETF Network Working Group RFC 2330*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2330.txt. 12

Rahrer, T., Faindra, R. & Wright, S. (2006). Triple-play Services Quality of Experience (QoE) Requirements. Technical Report TR-126, DSL-Forum, Architecture & Transport Working Group. 31, 112

Rigney, C. (2000). RADIUS Accounting. *IETF Network Working Group RFC 2866*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2866.txt. 27

Rigney, C., Willens, S., Rubens, A. & Simpson, W. (2000). Remote Authentication Dial In User Service. *IETF Network Working Group RFC 2865*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2865.txt. 27

Roberts, J., Mocci, U. & Virtamo, J. (1996). Broadband Network Teletraffic. In *Final Report of Action COST 242*, vol. 2. 35, 42, 62, 63, 108

Rosen, E., Viswanathan, A. & Callon, R. (2001). Multiprotocol Label Switching Architecture. *IETF Network Working Group RFC 3031*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3031.txt. 24

Sadasivan, G., Brownlee, N., Claise, B. & Quittek, J. (2006). Architecture for IP Flow Information Export. *IETF IP Flow Information Export WG - Internet-Draft*, available: 08-10-2008, online at: http://www.ietf.org/internet-drafts/drafts-ietf-ipfix-architecture-12.txt. 15, 79, 86, 90

Shenker, S., Partfidge, C. & Guerin, R. (1997a). Specification of Guarenteed Quality of Service. *IETF Network Working Group RFC 2212*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2212.txt. 40

Shenker, S., Partridge, C. & Guerin, R. (1997b). Specification of Guaranteed Quality of Service. *IETF Network Working Group RFC 2212*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2212.txt. 20

Simonian, A. & Guibert, J. (1995). Large deviations approximation for fluid queues fed by a large number of on/off sources. In *IEEE Journal on Selected Areas in Communicaions*, vol. 13, 1017–1027. 34, 68

Sinha, R., Papadopoulos, C. & Heidemann, J. (2007). Internet Packet Size Distributions: Some Observations. Technical Report ISI-TR-2007-643, University of Southern California, availavle: 09-10-2008, online: http://netweb.usc.edu/ rsinha/pkt-sizes/. 60

Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Zhang, M. & Paxson, V. (2000). Stream Control Transmission Protocol. *IETF Network Working Group RFC 2960*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2960.txt. 28

Tasgetiren, M.F., Sevkli, M., Liang, Y.C. & Gencyilmaz, G. (2004). Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem. In *LNCS Ant Colony, Optimization and Swarm Intelligence*, vol. 3172, 382–389. 142

Taylor, H.M. & Karlin, S. (1998). *An introduction to stochastic modeling*. Academic Press, 3rd edn. 168

TM-Forum (2005). Enhanced Telecom Operations Map (eTOM), The Business Process Framework. *TeleManagement Forum*, available: 09-10-2008, online: http://www.tmforum.org/. 31

Turán, Z., Veres, A. & Oláh, A. (1998). A family of measurement-based admission control algorithms. *In Proc. of the IFIP TC6/WG6.3 7th International Conference on Performance of Information and Communication Systems (PICS)*, 153–164. 40, 41

Vetter, P., Fredricx, F., Borgne, M., Bergmann, J., Areizaga, E., Adams, P., Renon, F., Mickelsson, H., Rajan, G., Gilon, E. *et al.* (2005). Multi Service Access Everywhere. In *Proc. of the 31st European Conference on Optical Communication (ECOC)*, vol. 5, 41 – 44. 112

Waldbusser, S. (1991). Remote Network Monitoring Management Information Base. *IETF Network Working Group RFC 1271*, online at: http://www.ietf.org/rfc/rfc1271.txt. 13

Waldbusser, S. (1995). Remote Network Monitoring Management Information Base. *IETF Network Working Group RFC 1757*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc1757.txt. 14

Wroclawski, J. (1997a). Specification of the Controlled Load Quality of Service. *IETF Network Working Group RFC 2211*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2211.txt. 20

Wroclawski, J. (1997b). The Use of RSVP with IETF Integrated Services. *IETF Network Working Group RFC 2210*, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2205.txt. 19

Yip, M.P., Marathe, M.V. & Hughes, D.A. (2005). Estimating equivalent bandwidth capacity for a network transport device using on-line measurements. *United States Patent*, **6891798**, available: 09-10-2008 ,online: http://www.freepatentonline.com/6891789.html. 143

Yu, H., Zheng, D., Zhao, B. & Zheng, W. (2006). Understanding User Behavior in Large Scale Video on Demand Systems. In *Proc. of the EuroSys conference*, vol. 40, 333–344, ACM Press, New York, NY, USA. 114

Zseby, T., Molina, M., Duffield, N., Niccolini, S. & Raspall, F. (2007). Sampling and Filtering Techniques for IP Packet Selection. *IETF Packet Sampling WG (psamp) - Internet-Draft*, online at: http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-10.txt. 17

List of Acronyms

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorisation and Accounting
AF	Assured Forwarding
AS	Autonomous Systems
ATM	Asynchronous Transfer Mode
ATSI	The Alliance for Telecommunications Industry Solutions
bps	bits per second
BTC	Bulk Transfer Capacity
DiffServ	Differentiated Services
DSCP	DiffServ Code Point
EF	Expedited Forwarding
FEC	Forwarding Equivalence Class
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IFF	IPTV Interoperability Forum
IMS	IP Multimedia Subsystem
IPDR	Internet Protocol Detail Record

IPPM	IP Performance Metrics
IPTV	Internet Protocol Television
ISO	International Organisation of Standardisation
ITU-T	International Telecommunications Union - Telecommunications Standardisation Sector
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multiprotocol Label Switching
PHB	Per Hop Behaviour
PSAMP	Packet Sampling
RADIUS	Remote Authentication Dial In User Service
RMON	Remote Network Monitoring Architecture
RSVP	Resource Reservation Protocol
RTFM	Real-time Traffic Flow Measurement
SNMP	Simple Network Management Protocol
SNMP	Simple Network Management Protocol
TACACS+	Terminal Access Controller Access-Control System Plus
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networking
TSpec	Traffic Specification
XML	eXtendible Mark-up Language

Appendix A

Appendix A

A.1 Queueing theory

Queueing theory is the mathematical study of queues or waiting lines. The theory enables mathematical analysis of several related processes, including arriving at the queue, waiting in the queue and being served by the server(s) at the front of the queue. The theory permits the derivation and calculation of several performance metrics including the average waiting time in the queue or the system, the expected number waiting or receiving service and the probability of encountering the system in certain states, such as empty, full, having an available server or having to wait a certain time to be served. Queueing theory is directly related to modelling the transportation of IP packets through a network. For further information on queueing theory see (Cooper, 1981).

A.1.1 First In First Out Queue

The term First IN First Out describes the principle of a queue processing technique. In other words what arrives at the queue first is handled first, what comes in next waits until the first is finished.

A.1.2 Priority Queue

The priority queue treats arrivals at the queue by ordering arrivals based on priority. With this processing technique, higher priority arrivals are served first while lower priority arrivals must wait in the queue until such a time as they are the highest priority.

A.1.2.1 Queue Modelling

Queueing models can be represented using Kendall's Notation.
A/B/S/K/N/Disc

- A is the interarrival time distribution
- B is the service time distribution
- S is the number of servers
- K is the system capacity
- N is the calling population
- Disc is the service discipline assumed

In general the following assumptions are made when modelling a queue system; $K = \infty$, $N = \infty$ and Disc = FIFO. Therefor the notation used becomes A/B/S. Some standard notations for distributions A and B include

- M for Markovian (exponential) distribution
- E_K for an Erlang distribution with K phases
- D for Degenerate (or Deterministic) distribution
- G for General distribution (arbitrary)
- PH for a Phase-type distribution

A.2 Probability distribution functions

In essence, a probability distribution describes a range of possible values that random variable can take on and the associated probability of that value being within some measurable subset of that range. The following define a number of useful probability distributions with their own properties. For more information on probability and statistics see (Mendenhall *et al.*, 2005).

A.2.1 Poisson Distribution

The Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time series the last event.

Parameters	$\lambda \in (0, \infty)$
Support	$k \in \{0, 1, 2, 3, ...\}$
Probability mass function	$\frac{e^{-\lambda}\lambda^k}{k!}$
Cumulative distribution function	$\frac{\Gamma(\lfloor k+1 \rfloor, \lambda)}{\lfloor k \rfloor !}$
Mean	λ
Median	usually considered $\lfloor \lambda + \frac{1}{3} - \frac{0.2}{\lambda}$
Mode	$\lfloor \lambda \rfloor$
Variance	λ
Moment generating function	$exp\lambda e^t - 1$
Characteristic function	$exp(\lambda(e^{it} - 1))$

Table A.1: Poisson distribution characteristics

A.2.2 Exponential Distribution

The exponential distributions are a class of continuous probability distributions. An exponential distribution arrises naturally when modelling the time between independent events that happen at a constant average rate.

Parameters	$\lambda > 0$ rate or inverse scale (real)
Support	$[0, \infty)$
Probability density function	$\lambda e^{-\lambda x}$
Cumulative distribution function	$1 - e^{-\lambda x}$
Mean	$\frac{1}{\lambda}$
Median	$\frac{\ln(2)}{\lambda}$
Variance	λ^{-2}

Table A.2: Exponential distribution characteristics

A.2.3 Gaussian Distribution

The Gaussian distribution (or normal distribution) defines a family of continuous probability distributions. Each distribution within the family can be defined using two parameters, location and shape; the mean and variance respectively. For example; the standard normal distribution has a mean of zero and a variance of one. The normal distribution can, quite effectively, be used to model the behaviour of many systems without a complete understanding of the underlying mechanisms. It is often called the bell curve because the graph of its probability density [1] resembles a bell.

A.2.4 Pareto Distribution

The Pareto Distribution is a power law [2] probability distribution that is commonly used to model observable phenomena. Originally the distribution was used to describe the allocation of wealth among individuals since it seemed to show rather well that a large proportion of wealth of any society is owned by a smaller percentage of the people in the society.

[1] A probability density function is a function that represents a probability distribution in terms of integrals

[2] A power law is any polynomial relationship that exhibits the property of scale invariance[3].

Parameters	μ location (real)
	$\sigma^2 > 0$ squared scale (real)
Support	$x \in \curvearrowleft$
Probability density function	$\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$
Cumulative distribution function	$\frac{1}{2}\left(1 + \operatorname{erf}\frac{1-\mu}{\sigma\sqrt{2}}\right)$
Mean	μ
Median	μ
Mode	μ
Variance	σ^2

Table A.3: Gaussian distribution characteristics

A.3 Markov Chains

A Markov chain is a discrete-time stochastic process with the Markov property. Having the Markov property means that, given the present state, future states are independent of the past states. In other words, the present state description fully captures all the information that can influence the future evolution of the process. At each time instant the system may change its state from the current state to another state, or remain in the same state, according to a certain probability distribution. The changes of the state are called transitions, and the probabilities associated with various state-changes are termed transition probabilities.

A Markov chain is a sequence of random variables $X_1, X_2, X_3, ...$ with the Markov property, namely that, given the present state, the future and past states are independent. Formally:

$$\Pr(X_{n+1} = x | X_n = x_n, ..., X_1 = x_1) = \Pr(X_{n+1} = x | X_n = x_n). \tag{A.1}$$

The possible values of X_i form a countable set S called the state space of the chain. Markov chains are often described by a direct graph, where the edges are labelled by the probabilities of going from one state to the other states. Markov chains are commonly used to model processes in queueing theory and how traffic is processed through a set

Parameters	$x_m > 0$ scale (real)
	$k > 0$ shape (real)
Support	$x \in [x_m; +\infty)$
Probability distribution function	$\frac{kx_m^k}{x^{k+1}}$
Cumulative distribution function	$1 - (\frac{x_m}{x})^k$
Mean	$\frac{kx_m}{k-1} fork > 1$
Median	$x_m \sqrt[k]{2}$
Mode	x_m
Variance	$\frac{x_m^2 k}{(k-1)^2(k-2)} fork > 2$

Table A.4: Pareto distribution characteristics

of queues. For further information on stochastic processes such as the Markov chain see (Taylor & Karlin, 1998).

A.4 Prediction and averaging

There are a number of approaches that can be used to predict a value within a sequence of events. The following approaches are commonly used:

A.4.1 Simple moving average

A simple moving average can be calculated over the previous n collected effective bandwidth measurements. Using a simple moving average to predict the future effective bandwidth level can lead to under provisioning of resources. Such an approach tends to smooth out traffic bursts making the admission control algorithm less responsive to changes in traffic intensity.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{A.2}$$

A.4.2 Weighted moving average

Using a weighted average policy to estimate future effective bandwidth levels can be useful, as key measurements can be prioritised over others. The difficulty is choosing appropriate values to assign weights to. A common approach is to use a simple scaled approach where the oldest value with be assigned $\frac{1}{n}$ and the latest value assigned a weight of $\frac{n-1}{n}$.

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{A.3}$$

A.4.3 Exponentially weighted moving average

The exponentially weighted moving average operates similar to the weighted moving average, however the weighting is based in an exponential function where the newest recorded value is assigned the highest weight, and all other values preceding this value are assigned exponentially decreasing weights. This approach can react better to fluctuations in bandwidth. EMA denotes the exponential moving average being calculated. α controls the memory of the exponentially weighted moving average function. N denotes the number of previous periods to be considered in the estimation. x denotes the current measurements to be included into the exponentially weighted moving average.

$$\alpha = \frac{2}{N+1} \tag{A.4}$$

$$EMA_i = EMA_{i-1} + \alpha(x - EMA_{i-1}) \tag{A.5}$$

A.4.4 Peak value

Using the peak value from the collected is by far the most conservative option. This will ensure no QoS violations will occur, as ample of bandwidth is being reserved for the admission on flows. On the other hand, this option is not suited to admission control traffic with a high degree of fluctuation. If, for example, there was a sudden drop in throughput, the peak measurement would not reflect this drop until the peak value is removed from the list of n measurements. From our perspective, we prioitise the importance of QoS violations therefore we choose this conservative approach to predicting future effective bandwidth levels for the proposed empirical admission control algorithm.

$$x_{peak} = MAX(x_1 \cdots x_N) \tag{A.6}$$

Appendix B

Appendix B

B.1 Further Packet Trace Details

The following figures have been generated from the packet traces collected from the MOME (MOME, 2008) and CAIDA (CAIDA, 2008) packet trace repositories. Details about the packet traces has been extracted from the associated repositories and entered into the tables below.

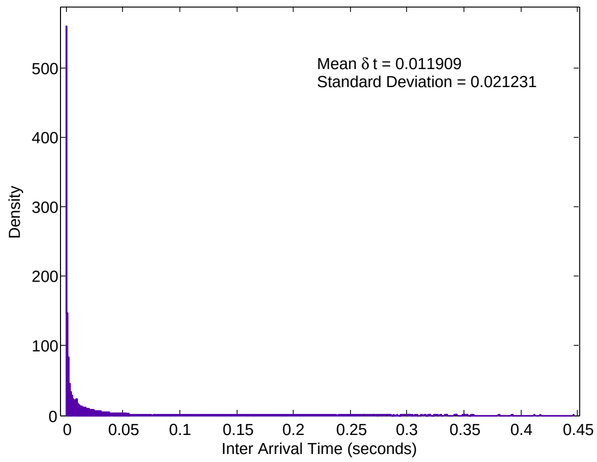

Figure B.1: Packet inter-arrival time distribution of MOME-1 packet trace.

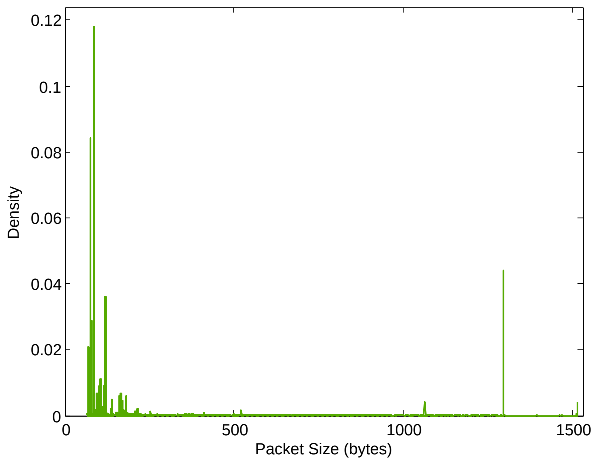

Figure B.2: Packet size distribution of MOME-1 packet trace.

Table B.1: Protocol breakdown of packet trace MOME-1.

Protocol		Packets	bytes	bytes/pkt
total		2000000 (100.00%)	486721373 (100.00%)	243.36
ip6		2000000 (100.00%)	486721373 (100.00%)	243.36
tcp6		1040336 (52.02%)	349547416 (71.82%)	335.99
	http(s)	14417 (0.72%)	15604641 (3.21%)	7082.38
	http(c)	399802 (19.99%)	47544669 (9.77%)	118.92
	squid	1043 (0.05%)	83179 (0.02%)	79.75
	smtp	5514 (0.28%)	1754845 (0.36%)	318.25
	nntp	295 (0.01%)	327942 (0.07%)	1111.67
	ftp	8448 (0.42%)	6889468 (1.42%)	815.51
	ssh	280614 (14.03%)	224319779 (46.09%)	799.39
	dns	197 (0.01%)	24008 (0.00%)	121.87
	bgp	81535 (4.08%)	7369096 (1.51%)	90.38
	realaud	1 (0.00%)	94 (0.00%)	94.00
	icecast	28 (0.00%)	2184 (0.00%)	78.00
	other	248442 (12.42%)	45627511 (9.37%)	183.65
udp6		446705 (22.34%)	68842277 (14.14%)	154.11
	dns	342044 (17.10%)	50095792 (10.29%)	146.46
	other	104661 (5.23%)	18746485 (3.85%)	179.12
icmp6		500670 (25.03%)	66875736 (13.74%)	133.57
ospf6		2381 (0.12%)	214290 (0.04%)	90.00
ip4		7675 (0.38%)	928474 (0.19%)	120.97
ip6		1435 (0.07%)	209330 (0.04%)	145.87
pim6		795 (0.04%)	101760 (0.02%) 7	128.00
other6		3 (0.00%)	2090 (0.00%)	696.67

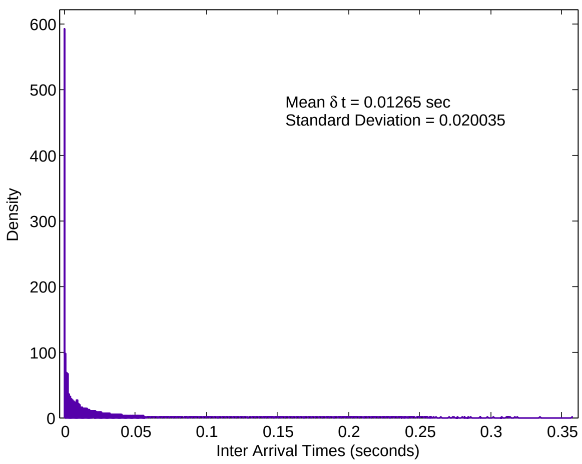

Figure B.3: Packet inter-arrival time distribution of MOME-2 packet trace.

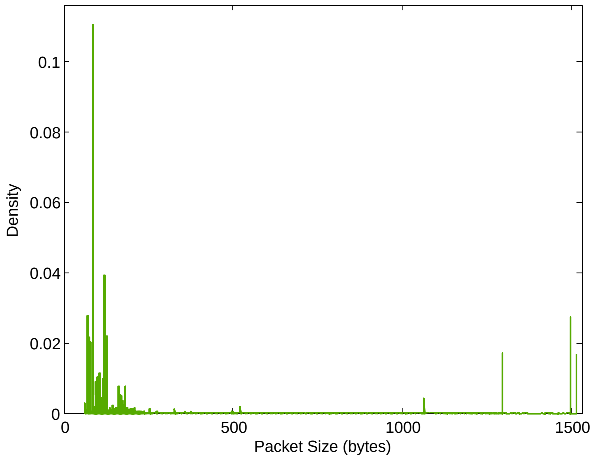

Figure B.4: Packet size distribution of MOME-2 packet trace.

Table B.2: Protocol breakdown of packet trace MOME-2.

Protocol		Packets	bytes	bytes/pkt
total		2000000 (100.00%)	590247636 (100.00%)	295.12
ip6		2000000 (100.00%)	590247636 (100.00%)	295.12
tcp6		698743 (34.94%)	379007724 (64.21%)	542.41
	http(s)	8200 (0.41%)	7766501 (1.32%)	947.13
	http(c)	55079 (2.75%)	5973530 (1.01%)	108.45
	squid	1182 (0.06%)	88852 (0.02%)	75.17
	smtp	13185 (0.66%)	2526839 (0.43%)	191.64
	nntp	1576 (0.08%)	148350 (0.03%)	94.13
	ftp	69045 (3.45%)	57025017 (9.66%)	825.91
	pop3	12 (0.00%)	936 (0.00%)	78.00
	ssh	13804 (0.69%)	2279670 (0.39%)	165.15
	dns	211 (0.01%)	24514 (0.00%)	116.18
	bgp	91109 (4.56%)	8455907 (1.43%)	92.81
	icecast	42 (0.00%)	3276 (0.00%)	78.00
	other	445298 (22.26%)	294714332 (49.93%)	661.84
udp6		585262 (29.26%)	92433977 (15.66%)	157.94
	dns	471196 (23.56%)	71413605 (12.10%)	151.56
	other	114066 (5.70%)	21020372 (3.56%)	184.28
icmp6		583158 (29.16%)	80288608 (13.60%)	137.68
ospf6		2529 (0.13%)	227610 (0.04%)	90.00
ip4		8285 (0.41%)	994052 (0.17%)	119.98
ip6		93802 (4.69%)	15050523 (2.55%)	160.45
ipsec6		121 (0.01%)	17666 (0.00%)	146.00
pim6		844 (0.04%)	108032 (0.02%)	128.00
other6		27256 (1.36%)	22119444 (3.75%)	811.54

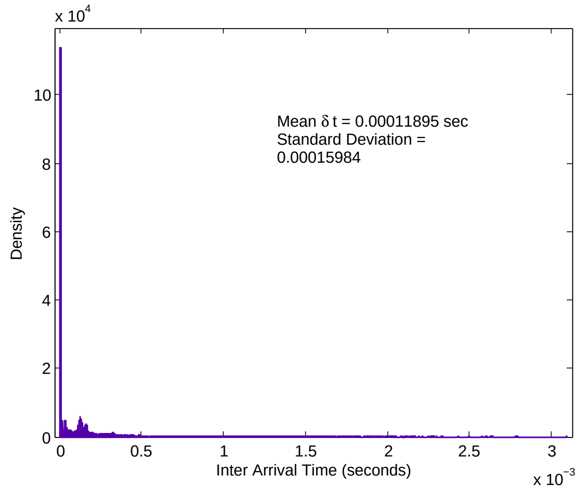

Figure B.5: Packet inter-arrival time distribution of MOME-3 packet trace.

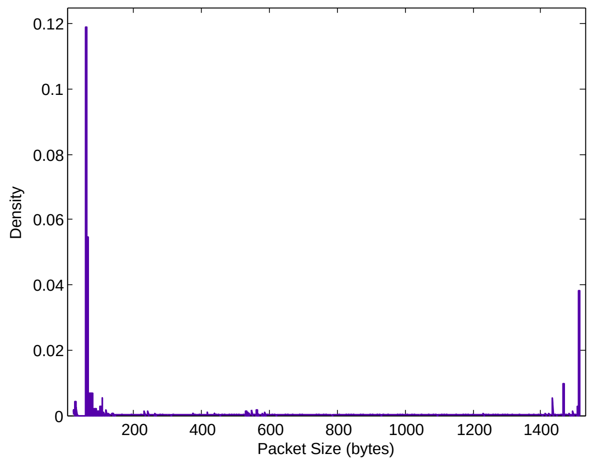

Figure B.6: Packet size distribution of MOME-3 packet trace.

Table B.3: Protocol breakdown of packet trace MOME-3.

Protocol		Packets	bytes	bytes/pkt
total		7564350 (100.00%)	2879403797 (100.00%)	380.65
ip		7394536 (97.76%)	2874285822 (99.82%)	388.70
tcp		6216875 (82.19%)	2614059063 (90.78%)	420.48
	http(s)	1196957 (15.82%)	1301819049 (45.21%)	1087.61
	http(c)	2383420 (31.51%)	304092581 (10.56%)	127.59
	squid	10681 (0.14%)	1168298 (0.04%)	109.38
	smtp	245957 (3.25%)	84519445 (2.94%)	343.64
	nntp	382 (0.01%)	36620 (0.00%)	95.86
	ftp	215697 (2.85%)	19251728 (0.67%)	89.25
	pop3	5230 (0.07%)	1309509 (0.05%)	250.38
	imap	4441 (0.06%)	1841297 (0.06%)	414.61
	telnet	1844 (0.02%)	220466 (0.01%)	119.56
	ssh	1539 (0.02%)	225783 (0.01%)	146.71
	dns	734 (0.01%)	55104 (0.00%)	75.07
	bgp	142 (0.00%)	51167 (0.00%)	360.33
	napster	1404 (0.02%)	147715 (0.01%)	105.21
	rtsp	37229 (0.49%)	6346783 (0.22%)	170.48
	icecast	27119 (0.36%)	32521285 (1.13%)	1199.21
	other	2057609 (27.20%)	859539435 (29.85%)	417.74
udp		794726 (10.51%)	199350283 (6.92%)	250.84
	dns	662449 (8.76%)	155567564 (5.40%)	234.84
	realaud	1468 (0.02%)	128360 (0.00%)	87.44
	halflif	3 (0.00%)	475 (0.00%)	158.33
	quake	993 (0.01%)	152818 (0.01%)	153.90
	other	121829 (1.61%)	42943560 (1.49%)	352.49
icmp		121913 (1.61%)	10202315 (0.35%)	83.69
ipsec		87 (0.00%)	8786 (0.00%)	100.99
ip6		533 (0.01%)	75604 (0.00%)	141.85
other		260402 (3.44%)	50589771 (1.76%)	194.28
frag		10444 (0.14%)	11740215 (0.41%)	1124.11

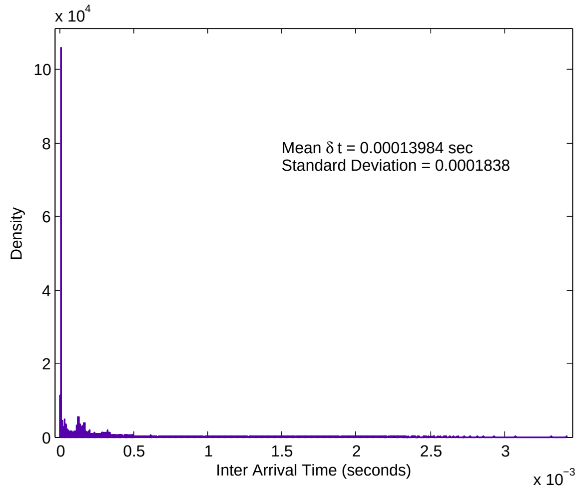

Figure B.7: Packet inter-arrival time distribution of MOME-4 packet trace.

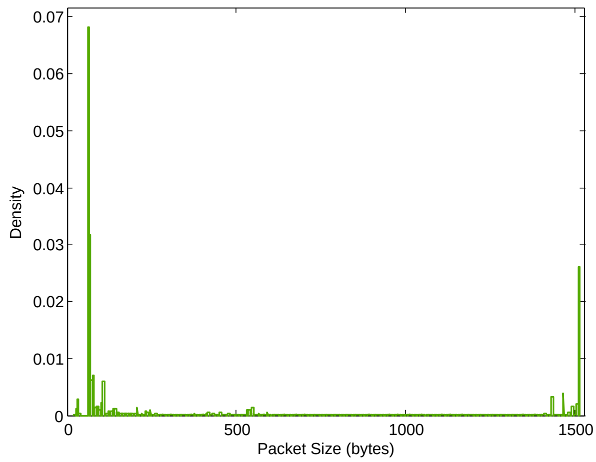

Figure B.8: Packet size distribution of MOME-4 packet trace.

Table B.4: Protocol breakdown of packet trace MOME-3.

Protocol		Packets	bytes	bytes/pkt
total		6434228 (100.00%)	2522747081 (100.00%)	392.08
ip		6319707 (98.22%)	2519304529 (99.86%)	398.64
tcp		4912250 (76.35%)	2020162001 (80.08%)	411.25
	http(s)	854955 (13.29%)	944343179 (37.43%)	1104.55
	http(c)	1838806 (28.58%)	227032214 (9.00%)	123.47
	squid	3088 (0.05%)	2499911 (0.10%)	809.56
	smtp	127477 (1.98%)	43629767 (1.73%)	342.26
	nntp	343 (0.01%)	31982 (0.00%)	93.24
	ftp	118210 (1.84%)	21701396 (0.86%)	183.58
	pop3	5636 (0.09%)	963784 (0.04%)	171.00
	imap	4374 (0.07%)	891584 (0.04%)	203.84
	telnet	1324 (0.02%)	112573 (0.00%)	85.02
	ssh	15001 (0.23%)	1114698 (0.04%)	74.31
	dns	1156 (0.02%)	102429 (0.00%)	88.61
	bgp	129 (0.00%)	36938 (0.00%)	286.34
	napster	23602 (0.37%)	14072673 (0.56%)	596.25
	realaud	27 (0.00%)	4165 (0.00%)	154.26
	rtsp	79334 (1.23%)	56468105 (2.24%)	711.78
	icecast	23830 (0.37%)	1661135 (0.07%)	69.71
	other	1798030 (27.94%)	704912012 (27.94%)	392.05
udp		1070028 (16.63%)	457286913 (18.13%)	427.36
	dns	613176 (9.53%)	148218549 (5.88%)	241.72
	realaud	5602 (0.09%)	1606602 (0.06%)	286.79
	other	417413 (6.49%)	280259232 (11.11%)	671.42
icmp		101129 (1.57%)	8699272 (0.34%)	86.02
ipsec		85 (0.00%)	7871 (0.00%)	92.60
ip6		447 (0.01%)	65455 (0.00%)	146.43
other		235768 (3.66%)	33083017 (1.31%)	140.32
frag		210691 (3.27%)	254094543 (10.07%)	1206.01

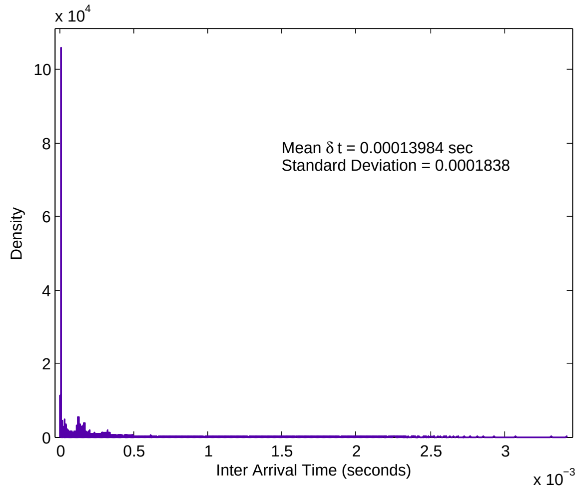

Figure B.9: Packet inter-arrival time distribution of CAIDA-1 packet trace.

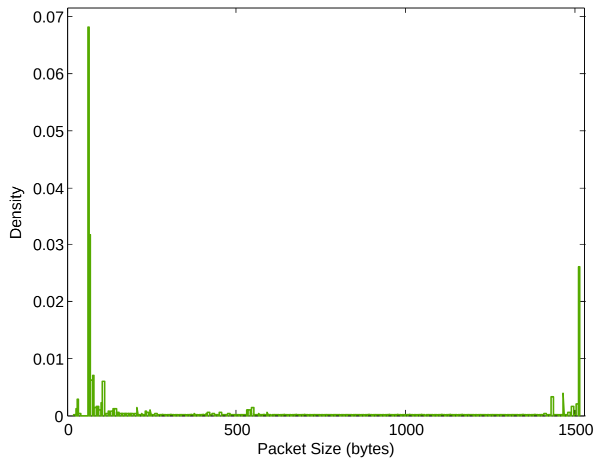

Figure B.10: Packet size distribution of CAIDA-1 packet trace.

Table B.5: Details of packet trace CAIDA-1.

Maximum capture length for interface 0:	unknown
First timestamp:	1210856347.126342000
Last timestamp:	1210856399.999997000
Unknown encapsulation:	0
IPv4 bytes:	5124699029
IPv4 pkts:	10106525
IPv4 flows:	758749
Unique IPv4 addresses:	375001
Unique IPv4 source addresses:	185279
Unique IPv4 destination addresses:	189723
Unique IPv4 TCP source ports:	48322
Unique IPv4 TCP destination ports:	47958
Unique IPv4 UDP source ports:	55223
Unique IPv4 UDP destination ports:	52651
Unique IPv4 ICMP type/codes:	15
IPv6 pkts:	491
IPv6 bytes:	95127
non-IP protocols:	0
non-IP pkts:	0