ABSTRACT

Title of Dissertation: PERFORMANCE MANAGEMENT IN ATM NETWORKS

Anubhav Arora, Doctor of Philosophy, 2002

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

ATM is representative of the connection-oriented resource provisioning class of protocols. The ATM network is expected to provide end-to-end QoS guarantees to connections in the form of bounds on delays, errors and/or losses. Performance management involves measurement of QoS parameters, and application of control measures (if required) to improve the QoS provided to connections, or to improve the resource utilization at switches. QoS provisioning is very important for real-time connections in which losses are irrecoverable and delays cause interruptions in service. QoS of connections on a node is a direct function of the queueing and scheduling on the switch. Most scheduling architectures provide static allocation of resources (scheduling priority, maximum buffer) at connection setup time. End-to-end bounds are obtainable for some schedulers, however these are precluded for heterogeneously composed networks. The resource allocation does not adapt to the QoS provided on connections in real time. In addition, mechanisms to measure the QoS of a connection in real-time are scarce.

In this thesis, a novel framework for performance management is proposed. It provides QoS guarantees to real time connections. It comprises of in-service QoS monitoring mechanisms, a hierarchical scheduling algorithm based on dynamic priorities that are adaptive to measurements, and methods to tune the schedulers at individual nodes based on the end-to-end measurements. Also, a novel scheduler is introduced for scheduling maximum delay sensitive traffic. The worst case analysis for the leaky bucket constrained traffic arrivals is presented for this scheduler. This scheduler is also implemented on a switch and its practical aspects are analyzed. In order to understand the implementability of complex scheduling mechanisms, a comprehensive survey of the state-of-the-art technology used in the industry is performed. The thesis also introduces a method of measuring the one-way delay and jitter in a connection using in-service monitoring by special cells.

PERFORMANCE MANAGEMENT IN ATM NETWORKS

by

Anubhav Arora

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2002

Advisory Committee:

Professor John S. Baras, Chair
Assistant Professor Richard La
Professor Armand Makowski
Assistant Professor S. Raghavan
Professor A. Udaya Shankar

DEDICATION

To my family.

ACKNOWLEDGEMENTS

The support and direction provided by my advisor Prof. Baras was invaluable in my research. Discussions with Prof. Tassiulas and Prof. Makowski helped in many aspects of the thesis. I am also grateful to Dr. Penafiel of Lucent Technologies for providing me an opportunity to implement a part of the protocol. Informal chats with Paritosh Tyagi, Vijay Bharadwaj, Sudhir Varma, Vineet Birmani, and Vipul Sharma were insightful into many aspects of networking. The support provided by my wife, Avani, was invaluable towards the completion of this thesis, as were the changes suggested by my father. I am also grateful for the support extended by my family since I have been away from home. Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Every distributed application has a set of requirements for the underlying network. The requirements usually stem from the expectations of the user of the application. For example, a user making a voice call over the network needs that the quality of transmission be acceptably good for the person at the other end, and the delay in transmission of speech low enough so that there is no perceived interruption in the conversation. Thus a voice connection is required to have very low losses and tightly bounded round trip delays. Alternatively, consider the transmission of digital media like music or movies over the network. In this case, the user expects a smooth audio or image that does not skip. If the image is a few seconds late, however, it is not bothersome to the user. Thus the application needs low losses, bounded (but not necessarily very tight) delay of data, and strictly bounded jitter in the delay in order that the receiver buffer never under-runs and the image never skips. Now consider the transmission of real-time digital video like news, stock quotes etc. In this case the user will expect a tighter delay bound in addition to the above requirements. Other applications like transfer of files over ftp or http do not require any kind of performance guarantees, and the flow control protocol usually can tolerate any delay or losses that the packets incur.

Hence the network requires technology which satisfies the varying require-

ments of distributed applications. Provisioning differentiated services to networked applications has been a subject of research and development for many years. The flip side of providing guarantees to applications is that the application is required to send traffic under negotiated constraints.

1.1 QoS: A Brief History

Last century was the age of technological innovation. The need to communicate fueled the rapid development and deployment of information exchange technology. The telephone was the first instrument to bring point-to-point voice communication into the daily lives of people. The underlying telephone network was a circuit switched network, in which bandwidth was assigned to a new circuit. This was a TDM (Time Division Multiplexed) network, with the source (the telephone) producing traffic of known bandwidth. Congestion in this network was defined as the rejection of a call, and was the result of unavailability of bandwidth at any of the nodes in the path of the call.

Advances in computer technology soon stimulated the need for computer-to-computer communication. This sparked the evolution of the data network. It was a packet switched network, meaning that there was no pre-assignment of bandwidth (connection-less) and packets were switched from node to node based on the source and destination information they carried. This is still the model for the IP (Internet Protocol) network that exists today. This network is assumed to be unreliable and the higher layer protocols (like TCP: Transmission Control Protocol) were designed with this consideration.

The cost of provisioning and maintaining numerous networks inclined the network providers to envision a new protocol to transport voice and data over the same underlying network. This led to the development of ATM (Asynchronous

Transfer Mode) networks that were Virtual Circuit switched. The bandwidth was pre-assigned to connections, however, the spare bandwidth (the difference of the provisioned and the actual bandwidth) was usable by other connections. This was achieved by statistical multiplexing of cells at every node. The cells of a connection always traverse the path provisioned at connection setup (connection-oriented).

In both IP and ATM networks, congestion is defined as the accumulation of packets (or cells) at the buffers of interfaces. Congestion results in delay and loss of packets. Quality of Service (QoS) is defined as the requirement of the user for the network in terms of measurable performance metrics. In IP networks, a router is unable to differentiate much between packets as it has no prior knowledge of the behavior and requirements of the user transmitting the packet. In ATM networks, the switch has knowledge of the behavior of the source (thus can police and reject cells from malicious sources) and the requirements of the source (therefore it can provide appropriate QoS to the cells). The concept of a flow in ATM networks introduces traffic engineering tools that provide differentiated QoS to connections.

The migration of the data network protocols towards flow based connection oriented paradigm was motivated by the need to provide differentiated QoS to flows. This need was a result of the growth of multi-media services in the network, especially real-time services including Voice over IP, Video Conferencing, Video on demand. The concept of Virtual Private Networks (VPN) introduced a logically separate network existing on the shared infrastructure of the internet. QoS provisioning in the internet is a must for deploying VPNs.

The ATM network however, was no panacea and the ubiquitous deployment of IP hosts and routers meant that ATM only served as a Wide Area Network (WAN) transport protocol for IP. IP, therefore still suffered from the deficiency of per-flow differentiated QoS provisioning. This led to the development of In-

tegrated Services Architecture (Int-Serv) and Differentiated Services Architecture (Diff-Serv). The basic principle behind these efforts was to provide the concepts of flow and resource provisioning in IP. These were quickly overtaken by the development of an altogether new transport protocol for IP called Multi Protocol Label Switching (MPLS). This protocol provided the means for explicitly provisioning resources for connections (like ATM) and also provisioning for IP flows identified at the edge of the network.

1.2 The Problem

In any connection oriented QoS provisioning protocol, at the connection setup phase a contract is defined between the network and the user introducing a flow in the network, that includes the QoS requirement of the user and also the behavior of the user traffic. The network provider may police the user traffic upon entry into the network and the network is expected to furnish the contracted QoS to the flow until the termination of the call. An important point to note is that the QoS requirements are always end-to-end in nature, i.e., the performance metrics are defined for the complete path of the call. Real-time connections demand strict control on the end-to-end delay and delay-jitter of the packets. In this thesis, the problem of QoS provisioning for real-time connections (providing guarantees on delay and jitter) in connection-oriented networks is considered.

Traffic Engineering deals with the control of data-flow inside the network and the issues of guaranteeing performance to connections. It is an important constituent of protocols like ATM, MPLS, and Diff-Serv. The principles of traffic engineering remain similar for these protocols. ATM is the classical representative of the concepts behind these protocols, with mature and precise standards defining

it. Henceforth, this thesis would be confined to the subject of traffic engineering for real-time flows in ATM networks.

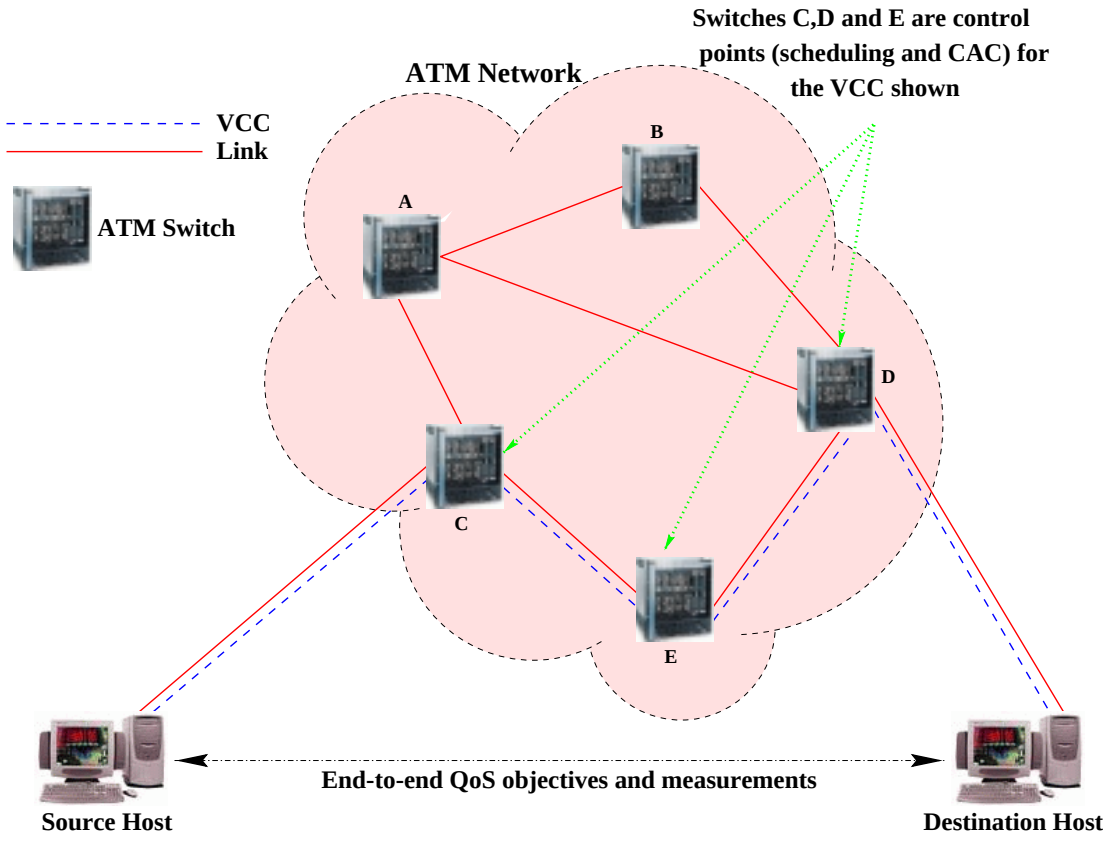

Figure 1.1: Network view showing the measurement and control points.

Specifically, the problem is to provide guaranteed end-to-end delay bound, delay-jitter bound, and a bound on the cell loss rate to a real-time connection on an ATM network. The standards pertaining to ATM [1, 2, 3] do not specify any mechanism for this problem, and the implementation is vendor specific. The algorithms that provide control over QoS on a network element are called the queueing and scheduling discipline. As will be made more definitive in the literature survey, a considerable amount of research has been dedicated to the study of queueing and scheduling in one network element. Though there are results to calculate bounds on end-to-end delays for certain schemes, there are hardly any mechanisms to pro-

vide the required end-to-end delays to connections. It is also important to note that due to the heterogenous nature of the network, network elements could be running different queueing and scheduling algorithms (figure 1.1). In this case the calculation of end-to-end delay bounds becomes exceedingly difficult.

Moreover, in every queueing and scheduling mechanism the resources are assigned at the setup phase to a connection and there is no procedure to change these resources in response to changing traffic conditions. In order to provide end-to-end QoS, it is also necessary for the network and the user to be able to monitor and measure the performance metrics in real-time connections. The standard pertaining to Operations and Maintenance in ATM [4] provides inadequate measures for monitoring, as will be observed in the next chapter.

1.3 The Approach and Contributions

The principal contribution of this thesis is to propose a traffic engineering scheme that can provide end-to-end delay guarantees to real-time connections, and also provide in-service measurements for the performance metrics. This is achievable even in networks which have different nodes running different schedulers (with some constraints). End-to-end guarantees are achieved by continuously measuring the end-to-end delay on the connection and correcting the resource provisioning parameters at the intermediate nodes. The function of QoS monitoring is performed by the switch at the edge of the network cloud. There are three distinct components of the framework: scheduling at switches to provide different types of QoS guarantees, adaptation of intermediate schedulers in response to end-to-end performance metrics, and mechanisms to perform in-service monitoring of end-to-end QoS metrics.

This thesis proposes a novel hierarchical scheduling discipline that provisions rate, delay and jitter bounds to connections. A new delay bounding scheduler (called Highest Relative Delay First (HRDF)) is proposed, analyzed and compared against the well known GPS scheduler. The schedulability region of HRDF is calculated, and it is smaller than that of GPS. However it is the same as the easily provisionable region of GPS. This scheduler is also implemented in an ATM switch and experiments are carried out to verify the behavior of the scheduler. A comprehensive review of the state-of-the-art of scheduling mechanisms in the industry is presented. The computational power of a few of the network processors commercially available leads to the conclusion that HRDF is implementable in hardware.

The thesis also presents schemes to perform accurate QoS monitoring. Specifically, mechanisms to measure the one-way delay and jitter using a special cell (called pattern cell) are proposed. Schemes to identify the bottleneck node in a path are also explored.

The subject of communicating the end-to-end metrics to the intermediate switches and initiating corrective measures in the scheduling parameters is left for further study. These control mechanisms and the monitoring schemes also need to be implemented in the prototype switch to understand the performance gain achievable with this framework in a real world scenario.

1.4 Organization of the Thesis

The next chapter starts with a brief tutorial on ATM and the mechanisms of scheduling and CAC that provide the essential traffic engineering tools. A survey of various scheduling mechanisms is presented next, followed by a review of schemes

in literature that attempt to provide end-to-end guarantees. The topic of QoS monitoring, the associated standard and some relevant papers on the topic are addressed next. This is followed by a review of some other papers relevant to this work. In chapter 3, the problem is re-introduced and a high level description of the performance management framework is presented. The scope of this thesis is also defined in this chapter.

Chapter 4 is devoted to the study of scheduling on one node. The proposed hierarchical scheduler is defined, and a study of the HRDF scheduler under leaky bucket flows follows. A comparison with the well-known GPS scheduling discipline is also presented. The implementation of the HRDF scheduler in an ATM switch is the topic of chapter 5. The platform and the modifications required are specified, followed by experimental results and their comparison with theory and simulations.

Chapter 6 presents a comprehensive review of the type of scheduling disciplines used by vendors manufacturing switches and network processors. This is used to establish the implementability of the HRDF scheduler. Monitoring of QoS is a subject of chapter 7, and a scheme to measure the delay and loss using in-service OAM cells is furnished. The conclusions and directions for future work are presented in the last chapter.

Chapter 2

Background and Literature Review

This chapter is composed of background material required for a complete understanding of the problem, and of work in literature and standards related to this research. Since the focus of this thesis is on proposing enhancements to ATM, a brief review of the ATM protocol is presented first. This section is a short summary of the definitions and mechanisms in various ATM standards [1, 2, 3]. The relevance and definition of the concept of scheduling is introduced in the next section. Section 2.2 is devoted to a review of the various scheduling disciplines proposed in literature. A few papers on the subject of end-to-end scheduling (provisioning end-to-end performance) are reviewed next. The topic of monitoring of performance metrics is explored in the next section, including a review of the pertinent standard [4]. Some other work related to the concepts examined in this thesis is documented in the last section.

2.1 ATM: A Review

ATM is a WAN protocol designed for high-speed transport of traffic from various kinds of traffic sources. It is a connection-oriented cell switching protocol. Overlaid on the physical link topology of links is a mesh of virtual paths amongst the various network nodes. A source node establishes a switched virtual channel,

connecting through various paths, to the destination using a signaling protocol. Information is then carried in fixed size cells (53 Bytes) on the virtual channel, the cells being switched at each intermediate node according to their path and channel identifier. Permanent virtual connections established by means other than signaling, for example by the network operator, are supported. ATM is designed to support various kinds of services, voice/video/web/data etc. using statistical multiplexing of cell traffic to achieve efficient resource utilization at switches and links [5]. VP switching, i.e., switching of cells based only on the path identifier is also supported on network nodes.

Every connection is characterized by its type:

Constant Bit Rate (CBR): used by applications like voice where the cell arrival rate is fairly constant over time and there are strict QoS requirements.

Real-time Variable Bit Rate (rt-VBR): used by applications like real time video, where the cell arrival rate is bursty due to the nature of the video coders and the QoS requirements are strict.

Non Real-time Variable Bit Rate (nrt-VBR): off-line video content, for example, in which the traffic rate is bursty but the QoS requirements on the delivery of the traffic are loose.

Unspecified Bit Rate (UBR): This type of traffic can have any type of cell arrival profile and is given no QoS guarantees.

The purpose of discriminating between connections in this way is to to provide efficient mechanisms of billing and to manage the traffic inside the network in a predictable manner. The traffic contract of the connection specifies its type and the various rate parameters associated with that type:

Peak Cell Rate (PCR): the maximum rate of arrival in any interval of time. This is used for CBR and VBR traffic types.

Sustained Cell Rate (SCR): the average rate of arrival, used by VBR connections. For CBR connections, SCR is the same as PCR.

Maximum Burst Size (MBS): The maximum number of cells a connection can send continuously at the PCR rate. Defined for VBR connections only.

Minimum Cell Rate (MCR): the minimum reserved bandwidth for a connection. This is occasionally used for the UBR traffic type.

The required QoS consists of the measures: Cell Transfer Delay (CTD), Cell Delay Variance (CDV), Cell Loss Ratio (CLR), Cell Error Ratio (CER). The parameters CTD, CDV, and CLR are important for real time connections in which delays are interruptions in service and losses may not be recoverable using retransmissions. The network provides statistical guarantees to the real-time connections in terms of the delay and loss parameters, for example, the fraction of cells incurring a delay more than CTD would be bounded, the jitter in the end-to-end delay would be bounded by CDV etc. The nrt-VBR traffic is usually rate based and requires guarantees on errors and losses.

An incoming call in an ATM network is first granted bandwidth from the source to the destination in the connection setup phase according to its type. Signaling messages are used to establish a Switched Virtual Circuit (SVC) across the network using a routing algorithm, or it may be done by other means (such as by the network administrator) in the case of Permanent Virtual Circuits (PVC). A switch on the path accepts or denies the request to support a call (call and connection will be used interchangeably). If the switch accepts the call it allocates resources to it, calculates the optimal next hop for the connection, and forwards

the request. Otherwise if a switch denies the request it sends the call back to the previous switch, which now tries to find a new path. The algorithm to accept or deny a new call at a switch is called Connection Admission Control (CAC).

The network provides a guaranteed bandwidth to the CBR service. Transmission at rates above the negotiated rate is violation of the traffic contract. For VBR service, the network guarantees an average rate and accommodates bursts of traffic to the best of its capability at the time. The VBR traffic contract defines an average rate (SCR), a peak rate (PCR) and a burst size (MBS). Maintaining average rate above the negotiated rate or transmitting bursts of more than MBS cells at PCR constitutes violation of traffic contract. Conformance with respect to the rate parameters of traffic contract is established at the network service access point using a traffic policer. The policer tags or drops any violating cells in accordance with the UPC (Usage Parameter Control) algorithms implementing the GCRA (Generic Cell Rate Algorithm) defined in [1]. Tagged cells are low priority cells and switches are at liberty to drop these cells in the event of congestion. The network employs a variety of traffic management functions (resource management at nodes, CAC, cell tagging, traffic shaping etc.) and congestion control functions (cell discarding, forward congestion indication etc.) to support the various kinds of service requirements [3], [1].

2.2 Scheduling and CAC in an ATM Switch

There are many switching and queueing architectures that have been proposed for ATM switches (packet switches in general). Short reviews of switching architectures are found in [6] and [7]. The architecture of many commercial switches today is output buffered (figure 2.1(a)). Output queueing offers the following advantages

over input queueing: the thoughput is maximized and the latency of cells is more controllable [8]. Although a considerable amount of research has been devoted to input-queued switching [9], it has not made its way into mainstream commercial switches. The focus of many input queued systems is to try to emulate output queueing using some intelligent algorithms.

Assumption 2.2.1 The focus in this thesis is on the output-queued switch architecture.

The fabric of the switch operates at speed equal to the sum of the maximum line rates of all the links that can be supported. Hence, the fabric is non-blocking and can transfer all the cells incoming on all input ports to their respective output port in any interval of time. Contention for the output line is the source of delays and queueing in such a design, as the ingress rate from the fabric can be potentially much larger than the line rate. At every output port, there is a buffer segmented on a per-type basis (CBR, VBR etc.). In most switches, the buffer is further segmented into per-VC queues (figure 2.1(b)). For every time slot on the output line the scheduling algorithm identifies the VC from the set of connections with non-empty queues whose head-of-the-line cell would be transmitted in that slot. The algorithm schedules the CBR and VBR streams to provide them the bandwidth negotiated at connection setup, and schedules the best-effort UBR streams in the remaining slots. The scheduling decisions in most algorithms are made on the basis of the rate requirement of a connection and not the desired QoS parameters. The set of all combinations of connections, such that if a particular combination is selected, it can be scheduled without any QoS violations, is defined as the schedulability region of the scheduling algorithm. The CAC would accept a call if the new load on the switch belongs to the schedulability region of the

scheduler. Since the scheduling typically is only a function of the rates of connections, CAC is also based on whether the desired rate can be accommodated by the scheduler [6], [7].

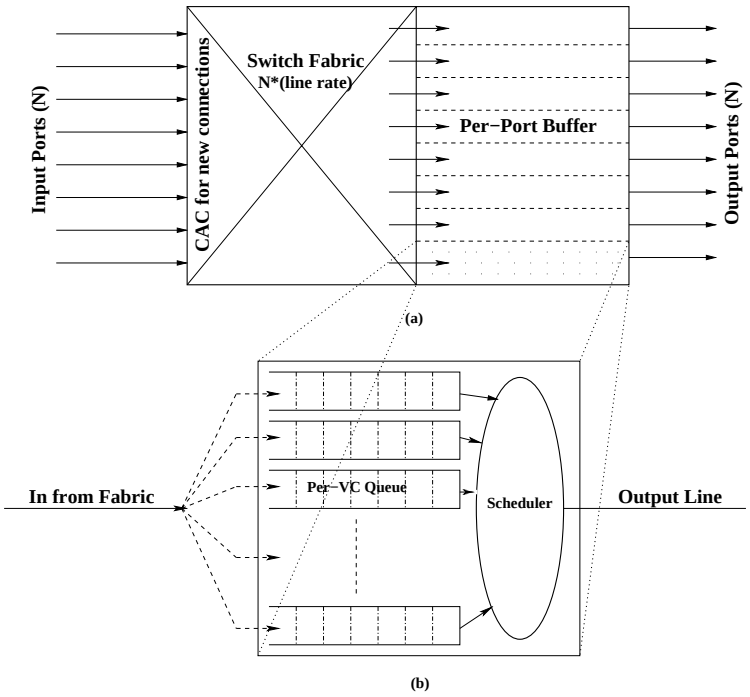

Figure 2.1: (a) Output buffered ATM Switch architecture (b) Per-VC Queued per-port buffer.

The QoS parameters: delay, jitter and loss are end-to-end performance parameters. Most of the scheduling algorithms in practice and in literature are designed to provide bandwidth guarantees to connections, usually optimizing on the the fairness of bandwidth distribution amongst connections. Most designs calculate

bounds on delay and loss in a node using a particular scheduling discipline and then provide the end-to-end bounds that can be supported using the knowledge of number of hops in a connection. Delay and bandwidth are thus very intimately related and this implies that a connection requiring a tight delay bound reserve a correspondingly large bandwidth regardless of its actual bandwidth requirement.

Another approach is to use traffic shapers (which buffer the incoming traffic and transmit it in a predictable pattern) at every node in conjunction with the scheduler to provide deterministic bounds, which are again a function of the shaping and scheduling discipline. Although traffic shapers provide predictable cell departure pattern at every node, the scheduling delay of these schedulers is larger because of their non-work conserving nature. Consequently, shapers are not very well suited for real-time traffic. The switches in the network are from different vendors and use varying traffic engineering implementations. In this case, it is very hard to guarantee end-to-end QoS. To separate QoS from bandwidth requires translation of end-to-end QoS requirements to bounds for each node. This has been ad-hoc, based on the knowledge of number of hops on the path which may not be known at the connection establishment phase. Thus at present the resource allocation at nodes (in the form of scheduling parameters) at connection setup does not incorporate the desired end-to-end QoS bounds in any of these approaches. Also, these methods are static and resources once allocated are not changed during the lifetime of a connection.

2.3 Scheduling Disciplines

The scheduling controls the delay of cells at a switch and the CAC assigns resources to new connections while providing protection to existing connections. New con-

nections that can potentially cause congestion in the switch are not accepted. The CAC algorithm is determined by the scheduling algorithm. In this section the different scheduling algorithms proposed for packet switches will be reviewed, with the viewpoint of identifying algorithms capable of providing per connection QoS at a single node.

The scheduling disciplines can be broadly divided in the following categories:

1. Priority scheduling
2. Work-conserving fair share scheduling - Generalized Processor Sharing (GPS) Based
3. Work-conserving fair share scheduling - Round Robin (RR) Based
4. Delay based scheduling
5. Traffic shaping schemes

2.3.1 Priority scheduling

A priority based scheduler assigns priorities to each queue and serves in the order of the priority. A lower priority queue would be served only when there are no cells waiting in the higher priority queues. There is usually FIFO queueing within one priority. The QoS of the lower priority queues is thus completely determined by the behavior of the higher priority queues. Applied to ATM, priorities are usually assigned equal in one traffic class for example, all CBR traffic enters only one queue. This kind of scheduling is QoS insensitive and does not take into account any delay or loss requirements of the real-time streams. Priority queueing is a very simple arbitration operation and it can provide a class of real time traffic, a class of non-real time traffic and a best effort class. Increasing the number of classes

may not provide sufficiently fine-grained QoS commitments unless some level of fair access to bandwidth is provided [6]. Static per-class priority queueing is thus not practical for use when the objective is to provide QoS guarantees to individual connections. The advantage of this scheme is that it is probably the easiest to implement amongst all schemes discussed later.

2.3.2 Work-conserving fair share scheduling - GPS based

A fair share scheduler guarantees a fair share (defined by implementation) of the link bandwidth to a queue according to its weight (reserved bandwidth). It introduces isolation between the queues based on their bandwidth demand. A work conserving scheduler is one which is never idle when cells are present in any of the queues. An ideal fair share scheduler employs Generalized Processor Sharing, in which the traffic is given service in ratio of their weights in infinitesimal service amounts. Thus at any instance of time, a GPS scheduler would identify the set of queues that are backlogged, and then service them in the ratio of their weights for an infinitesimal time. Subsequently, it revisits the evaluation of the backlogged set and continues in this fashion. The basic unit of service is a cell in an ATM switch and so there exist various approximations of GPS (like WFQ, PGPS) in which service is given in finite increments.

Fair Queueing (FQ) [10]

Each cell is assigned a service deadline which is the finishing time of the cell if the server was a bit-by-bit round robin (BR) server. The cells are served in increasing order of finishing times, and the system emulates a BR server. The finish time of cell i of connection j is assigned as

$$F_i^j = max[F_{i-1}^j, R_i^j] + 1$$

where R_i^j is the number of rounds made. This assigns equal bandwidth to all connections. In Weighted Fair Queueing (WFQ) , bandwidth is assigned to connections in proportion to their weights.

Packetized Generalized Processor Sharing (PGPS) [11, 12]

PGPS is the generalized version of the FQ algorithms that assigns weights to a connection and virtual finish times to cells. The weight allows each connection to get a weighted share of the bandwidth. PGPS approximates GPS to within one cell time. The virtual time of the switch here is calculated according to the total service provided by the switch to all back-logged connections. This computation of the virtual time (also called the system potential) is very complex and is usually a computational bottleneck. The next two schemes keep track of the virtual time in a simpler way and are similar to PGPS otherwise.

Self Clocked Fair Queueing (SFQ) [13]

A more practical scheme than PGPS which keeps track of the virtual time in a switch as the service tag of the cell that is in service. It is reset to zero at the end of a busy period.

Worst-Case Fair Weighted Fair Queueing (WF^2Q) [14]

In this discipline, the next cell chosen to serve is the one that would have completed service in the corresponding GPS system at time t. It is shown that WF^2Q differs by not more than a cell from GPS. It also uses a lower complexity virtual time function than PGPS.

Summary

The generic feature of all these schemes is that each connection is assigned a weight and the server guarantees the weighted proportion of the bandwidth to the connection. This guarantee is over a large period of time, and the scheduler cannot be cognizant of any independent delay requirement of the connections. However upper bounds to the delay can be computed in all of these schemes. This introduces tight coupling between the bandwidth requirement and the delay guarantee provided. Also in general the computation of the delay bound is a very involved operation. These schemes are thus not suitable for scheduling QoS sensitive connections but are highly desirable for bandwidth provisioning schedulers.

2.3.3 Work-conserving fair share scheduling - RR based

All the schemes in the previous section were based on virtual finish times. It requires keeping track of a virtual time function in the scheduler and also timestamping each cell with a deadline (or finish time). This considerably complicates the design of the switch and introduces higher computational complexity. The round robin schemes however can be built using frames that do not require a timestamp on every cell, and without the need for heavy real-time computations (required computations may be done off-line, or not so often as every cell time).

Weighted Round Robin (WRR)

This is the simplest round robin scheme, with time divided in frames. Each frame transmits N cells and connections get bandwidth in proportion of their weights. The function giving the mapping from time slots to connections is executed whenever CAC is initiated and thus it is quasi-static. The bandwidth granularity that

can be provided is the N^{th} portion of link speed. Increasing N increases the granularity but also increases the latency of the connections. The scheduler's behavior is also dependent on its behavior when the queue which is scheduled at the current time slot has no cells to send.

Deficit Round Robin (DRR) [15]

This is an extension of the round robin scheduler. It maintains a deficit counter for each connection, according to the difference in the number of cells that a connection could have transmitted in the frame and the number of cells it actually transmitted in that frame. It then gives credit to connections based on the counter in subsequent rounds.

Uniform Round Robin (URR) [16]

It is a round robin discipline that tries to provide the fair share of a connection in a uniform manner, i.e., the slots of a connection in a frame would be uniformly distributed rather than being in a contiguous block (over a period of time). Thus a compromise on a complex slot scheduling function provides better distribution of the delay of cells.

Summary

All of these schemes are again insensitive to the QoS requirements of real-time connections. The distribution of delay of cells in these schemes is actually worse than in the GPS based schemes, although the computational effort required is significantly less. Therefore they are again unsuitable for the QoS provisioning schedulers and can be used for bandwidth provisioning.

2.3.4 Delay bounds based scheduling

The schedulers described earlier were concerned with only providing a fair share of bandwidth to a queue. Now some schemes for the optimization of specific parameters like CTD and CDV are discussed.

Earliest Deadline First (EDF) [17]

At connection setup, a maximum delay bound is fixed for every connection. Every incoming cell is stamped with a deadline (ingress time + the delay bound). The egress scheduler serves the cells in the increasing order of the deadlines. It is has been showed that for traffic bounded by leaky bucket algorithms (as in the case of ATM), the schedulability region of EDD is the superset of schedulability regions of all scheduling disciplines.

Rotating Priority Queue+ (RPQ+) [18]

This scheduler uses a set of FIFO queues with dynamically changing priorities. The order of the priorities is rotated periodically to increase the priority of waiting cells. The authors prove that this scheduler approaches the EDD scheduler in the limiting case of the rotation period tending to zero. Thus this is an approximate EDF implementation with much lower computational complexity.

Delay Earliest Due Date (D-EDD) [19]

A deadline is assigned to a cell according to its ingress time at the switch and a maximum delay parameter agreed upon at connection setup. Two kinds of services are provided - Deterministic bounds and Statistical bounds (lower priority). Both the queues are sorted according to the deadlines of the cells. Cells from deterministic queue receive higher priority, in order to provide deterministic delay bounds.

Consequently, the lower priority cells get statistical bounds.

Jitter Earliest Due Date (J-EDD) [20]

It is an extension of D-EDD to provide better jitter bounds. Essentially each node in the network has to preserve the arrival pattern of cells in a connection to the output line. In this scheme, a cell is stamped with the correction term - the difference in the time it was served and it was supposed to be served. A traffic shaper at the next switch would buffer it for that much time, hence preserving the cell generation pattern end-to-end. The disadvantage here is that for ATM, there is no space in the traffic cells to store the delay correction term.

Variation Fluctuation Smoothing [21]

This algorithm estimates the clock of each connection by using on-line measurements. It then computes the lateness of the head-of-the-line cell of each connection and transmits the latest cell. It is shown that the VFS scheme outperforms FIFO and EDD in providing reduced jitter for CBR traffic.

2.3.5 Traffic shaping

Deterministic Delay guarantees using Traffic Shaping [22]: This uses a traffic shaper in conjunction with a rate monotonic priority scheduler. The traffic is first shaped at network access and then at every switch using a leaky bucket. The scheduler is based on non-preemptive static priorities and provides the bucket drain rate to the queues.

2.3.6 Discussion

From the above review, it is observable that the scheduling schemes cater to very specific objectives, for example, fair allocation of bandwidth or maximum delay guarantees or bounds on CDV etc. Each of them is suitable only to specific class of service in order to optimize different parameters. Also the disciplines either tightly couple delay and bandwidth together or look at only delay requirements, and in order to provide end-to-end guarantees, a homogenous composition of the network needs to be assumed. However in a real ATM network, there are multiple classes of traffic on a switch with many widely different QoS objectives ranging from a fair share of bandwidth to delay and loss guarantees to both together. A modular scheduling architecture, components of which optimize different parameters like maximum delay, delay variation and fairness of distribution of bandwidth is thus needed. A higher layer arbitration policy to assign resources to these modules is also required. Such a scheme would be presented later in the thesis as an integral part of the performance management framework.

The ultimate goal of performance monitoring is to provide end-to-end QoS guarantees. All the schedulers reviewed above provide guarantees at only one node. In the next section some schemes for end-to-end provisioning are looked at.

2.4 End-to-end Scheduling

A few papers that propose distributed scheduling algorithms are discussed. Distributed scheduling tries to guarantee end-to-end performance.

Distributed scheduling

In [23], the authors propose a coordinated scheduling scheme to guarantee end-to-end delay bounds. The deadline of a cell passing through multiple switches running D-EDD scheduling is iteratively computed at each switch (using the previous deadline stamped on the cell). The deadline at the first switch is kept random. The analogy for this scheme is that of co-ordinated traffic lights which turn green after suitable intervals to allow traffic to pass uninterrupted once it has stopped at one light. This is not suitable for ATM as it requires stamping cells with a parameter and also provides only delay guarantees.

QoS driven scheduling

In [24], the authors redefine fairness as the proportion of customers receiving poor QoS in times of congestion rather than all customers getting a fair share of bandwidth. They present a scheduling strategy based on dual queues that allows the service provider to select its QoS objectives. Although they show their technique to be superior in scalability and QoS for real-time applications than fair queueing, the model used is of two FIFO queues. Thus it cannot be implemented in per-VC queueing architectures and networks where many different types of services exist and wide ranges of QoS levels need to be guaranteed.

Summary

It can be observed that the distributed scheduling schemes optimize for a single QoS requirement or a single class of service, and that too by assuming the same scheduler implemented on all nodes. ATM is composed of wide variety of classes of service and a range of QoS requirements which these schemes do not address. None of the schemes attempt to provide a broad spectrum of QoS to connections,

and none of the schemes assumes a network with diverse switches in it.

2.5 QoS Monitoring

The Operations, Administration and Maintenance (OAM) standard for ATM from ITU-T (International Telecommunication Union - Telecommunication) [4] specifies fault management and in-service performance monitoring mechanisms. The standard defines performance monitoring as "A function which processes user information to produce maintenance information specific to user information. This maintenance information is added to the user information at the source of a connection/link and extracted at the sink of a connection/link. Analysis of the maintenance event information at the sink of the connection allows estimation of the transport integrity to be analyzed". In this section, the procedures and algorithms in the OAM standard, and then some papers on QoS measurements are reviewed.

2.5.1 The OAM Standard for ATM

The OAM standard for ATM is recommendation I.610 from ITU-T [4]. Performance monitoring is performed by inserting end-to-end or segment (a segment defined as one or more link) monitoring OAM cells at F4 (VP) or F5 (VC) level. These cells monitor a block of user cells (block can be of size 128, 256, 512 or 1024 cells). The main objective of this monitoring is to detect errored blocks and loss/mis-insertion of cells in a block. There are forward monitoring cells inserted by the source to measure parameters along the connection and backward reporting cells inserted by the destination in the back-channel to report the measured values to the source. The various fields defined for these cells include

1. A sequence number to identify the cells

2. The total number of user cells sent (a modulo 64k counter) and the number of user cells with high Cell Loss Priority (CLP = 1)

3. A block error detection code (even parity/Bit Interleaved Parity - 16)

4. An optional time stamp for delay measurements

5. For backward monitoring cells: total number of user cells received, number of user cells with CLP=1 and the block error result on the received cells

Using these fields, block errors and differences in the number of transmitted and received cells can be determined. However, this does not give precise information about the number of lost and the number of mis-inserted cells. One way delay measurement requires that the clocks at the source and destination be synchronized, thus only round trip delays are accurately measurable using the optional timestamp field and loopback of cells at the destination. Moreover, the timestamp field is optional and is not implemented by most applications currently. Round trip delay measurements do not provide the capability to pinpoint the bottleneck links/segments in the circuit as well. The standard at no point mentions the objective of using the minimum overhead of OAM cells or any related algorithms. The precision with which the measurements are to be made, both the precision of each measurement and the interval between measurements, is not addressed in the standard. The question of verification of QoS when the total number of cells transferred is small has also not been addressed in [4]. Thus it is observable that the OAM standard for ATM does not specify many of the performance monitoring objectives.

2.5.2 Delay measurement using management cells

The authors of the papers [25, 26] propose that the one-way cell delay can be accurately measured by segmenting it into delays experienced at each switch. A switch can time-stamp a management cell when it is received at its input. At the departure of the cell, the cell delay can be computed and added to the delay value already written in the cell (initialized to zero by the source). The processing delay required for this design and the propagation delays are fixed and can be precomputed. Thus, a management cell accumulates the delay along the path. The delay field at the destination gives a sample of the cell transfer delay which does not suffer from the clock synchronization problem, as the differences are taken from the same clocks. An alternative is to have multiple time-stamp fields in the management cell so that the delay at each switch can be recorded separately. This may be useful for diagnostic purposes, for example to determine the bottleneck link.

Management cells are required to occupy a small portion of the bandwidth, especially when the network is congested. This criterion dictates the inter-sample time. The authors of [25] have briefly alluded to this issue and stated some heuristic arguments.

This scheme requires new processing capabilities at the switches to modify cells on ingress and egress. No mention is made about the accuracy of such a scheme either (which may or may not justify the additional complexity in the switches). The next section describes a procedure based on statistical analysis of the remote clock which can be used with OAM performance management cells for better estimation of delay parameters.

2.5.3 Delay measurement using clock parameter estimation

The technique proposed by Roppel in [27] relies on estimation of the one-way cell transfer delay by analyzing the properties of the remote clock. Essentially, the remote clock can be modeled, its parameters estimated, and the time-stamp of the destination can be corrected for the offset. In this method, the switches do not need any new processing capabilities.

The remote clock $C(t)$ can be modeled using a time offset parameter (ΔT_o) and a clock frequency drift parameter (α):

$$C(t) = t + \Delta T_o + \alpha(t - t_o) + \epsilon(t).$$

These two parameters can be estimated using a regression model on the delay samples from the backward reporting performance management cells. This method however requires that the minimum delay along both directions to be the same. The number of samples required to converge to correct clock parameters may be large, thus the time for convergence for low bit rate links can be very high. These disadvantages can prohibit the use of this scheme for a large class of networks and connections.

2.6 Other Related Work

2.6.1 Traffic measurements

Actual traffic measurements are made by the authors of [28] on the vBNS network to observe the effects of traffic going through various hops, on the QoS parameters and peak / sustained rate requirements of the stream at every node. The main results presented therein are that the standard probability distributions do not

fit delay and loss parameters of streams traveling multiple hops and the PCR requirement of a CBR stream increases at every hop in proportion to its rate due to increased burstiness. The experiments indicate that the resource allocation to a stream should not be the same at every switch and the scheduling and CAC algorithms should be distributed in nature.

2.6.2 Multi-agent based intelligent Network Management model

In [29], a distributed management model using semi-autonomous agents making local decisions as far as possible and communicating with other agents using a blackboard is defined. The agents try to co-operatively converge to a solution on the blackboard, which is controlled by a central controller.

The model defined in the paper is very high level and a lot of details on implementation, algorithms, usefulness in a multi-service network have not been considered. The simulation results are for congestion problems in very simple network topologies, and very elementary traffic sources.

2.6.3 Performance Monitoring model

In [30], the author proposes a general model for performance management in ATM networks based on interaction between management and control functions. The paper discusses the functionality of the Network Management System, the plane & layer management in switching systems, the call control functions and that of QoS management from the perspective of a service platform. The paper then goes on to define performance monitoring in ATM networks and illustrates the use of OAM cells [4] for in-service performance measurements. Overall the paper brings out a number of issues and highlights the interaction between different systems

but does not present any solution to any of the problems.

2.6.4 CLR estimation for applications to CAC

A delay measurement based CAC algorithm, that can provide predictive service (allowing occasional violations in QoS) is proposed in [31]. The delay being measured and guaranteed is local switch delay and the problem of end-to-end delays is not addressed here. In [32], the authors propose the use of measurement of buffer occupancy and inferring the QoS parameters using an approximating functions. These measurements are then used in the CAC algorithm. In [33], the authors propose measuring CLR for small systems and using a fuzzy algorithm in conjunction with the knowledge of asymptotic behavior of large systems to predict the CLR for large switches. No source models are assumed here and the CAC algorithm solely depends on the predicted and demanded CLR and the rate of a connection.

Chapter 3

A Framework For Performance Management

3.1 The Problem Revisited

In ATM, the application requirements are mapped to various traffic classes and their different attributes. The CBR class of service receives the highest priority in turn providing the best possible delay and loss performance. However the traffic contract dictates the maximum rate that the application can feed to the network. Voice calls are typically carried over this class, and since the voice applications are usually TDM based, the rate fed into the network is constant and known a-priori. Real time video is usually fed to the next class of service: rt-VBR. The coding of the video source is such that the maximum and the average rates of data can be very different. Ideally, both of these applications should be capable of informing the network about their delay and loss requirements in quantitative terms. These requirements of the applications are always end-to-end in nature, i.e., the applications expect the QoS from the source to the destination regardless of the number and type of hops in the path. As noted from the literature review, the research on this subject is considerably less than the research on QoS guarantees on a single node. On this subject, Keshav, in his recent book on Computer Networking [7], comments "In a heterogenous network, where different parts of

the network may employ different scheduling disciplines, guaranteeing end-to-end performance bounds is a hard problem, and an area of active research". Further, these requirements last for the lifetime of the connection, and thus the network needs to provision the stipulated QoS for this time.

The end-to-end application requirements are global requirements that need to be translated into a set of local requirements for the purpose of resource reservation on the network elements (referred to as "resource translation" in literature). There are some heuristic methods to accomplish this during signaling. For example: in the forward pass of the setup, resources are reserved in a conservative manner. The destination compares the requested bound with the achievable quality and resources are relaxed in the reverse path to reserve only as much as required. In Keshav's words, "Resource translation is an area where we still need research into good heuristics or optimal algorithms". Howsoever this may be accomplished, connections are allocated resources once at the connection setup phase in every protocol and these resources are not changed until the connection is terminated.

The network establishes the conformance of the traffic with respect to the contract at the point of ingress. As the connection traverses along various hops, the burstiness characteristic of the traffic keeps changing (due to the burstiness introduced by switches) and may not conform to the contract [28]. These inaccuracies can easily lead to transient congestion in the network elements. This in turn means that the QoS provided to a connection can possess a wide spectrum during its lifetime, especially in light of ad-hoc resource translation algorithms and the heterogenous nature of the network elements. This suggests that for real time connections, there should be methods to correct the resource allocation of the calls in response to the QoS provided, in order that the network can fulfill its portion of the traffic contract. For this to be achievable, the QoS provided to a connection

needs to be known in real time, i.e., the in-service quantitative measurement of QoS of a connection should be possible. As observed from the literature review section, the mechanism specified in the standard [4] to measure the delay and jitter of a cell stream is inadequate and inaccurate.

Therefore a multiservice network like ATM requires a complete framework for performance management to provide end-to-end QoS provisioning for real time connections. The following are some of the requirements for the design of such a framework:

1. Need tools to guarantee various QoS metrics from the source to the destination.

2. Mechanisms to change the resource allocation during the life of a connection should be present.

3. Means to accurately measure the QoS of real time connections are needed.

The provisioning of QoS to the cells of a connection on a single node, given the knowledge of the required performance bounds, is not an easy task either. The requirements imposed vary widely in nature from bandwidth guarantees to bounds on delay, jitter and loss. There is a significant body of literature devoted to the subject but as noted from the previous chapter, the design of each scheduling discipline is to optimize on a single objective. For example, most schedulers reserve bandwidth and guarantee delay by calculating the delay bound given the bandwidth reservation. As an integral component of the framework, a multi-functional scheduling discipline is required that can provide different types of bounds. Giroux and Ganti allude to this aspect in their book on QoS in ATM [6]: "It may be more efficient to use a combination of scheduling mechanisms [in a network node]."

With reference to QoS monitoring Chen et. al. [25] discuss the use of special cells for a variety of issues, among them being the in-service measurement of delay and loss. They propose a technique based upon timestamping the cell at the ingress of a switch and then stamping the delay of the cell in the switch on at the egress of the cell. A major flaw in this technique is that this is practical only in a completely output queued switch. In a combined input-output queued switch, the ingress and egress of a particular cell in the switch are typically at different line cards of the system and it is not practical to maintain a synchronized high precision clock at all line cards. Thus the timestamp put on the cell by the ingress line card is meaningless to the egress line card and the delay on the switch is not known. Even though this paper brought out a number of issues in management and control of ATM networks, to the best of our knowledge there is no subsequent work employing the same concept of using special cells for the purpose.

A novel framework for performance management is proposed int the next section. It is designed to address all the problems and concerns related to end-to-end provisioning mentioned above.

3.2 High-Level Description

3.2.1 Requirements

The proposed framework for performance management in ATM networks builds upon the discussion on QoS provisioning in the previous section. The essential theme is to perform in-service QoS monitoring for real-time connections, and correct the resource allocation parameters at the intermediate nodes if required. This segments to the following three major requirements from the design of the network:

1. ability to monitor the QoS of a real-time connection, using the in-service monitoring methods to measure the end-to-end delay, jitter and loss

2. flexible QoS provisioning at intermediate nodes, (i.e., correctable scheduling and CAC parameters on switches)

3. algorithms to regulate the QoS of a connection in real-time in order to either improve the QoS, or to improve the resource utilization at nodes.

3.2.2 Assumptions

Assumption 3.2.1 When a connection is handled by multiple network service providers, it is assumed that suitable mechanisms exist to delegate QoS requirements to each of the service providers (beyond the scope of this work). Each service provider then monitors and controls QoS that it is obligated to provide in its own domain. In this thesis, the provisioning of QoS in one domain would be considered.

Assumption 3.2.2 It is assumed that every connection is policed for its UPC contract parameters at the network access points, and is conforming to its traffic contract once it has entered the network.

3.2.3 Monitoring

QoS Monitoring of a connection is accomplished using specialized devices for injecting and extracting OAM cells (called "Performance Management (PM) device") placed immediately prior to the network access equipment (figure 3.1) by the service provider. RadComm is one of the vendors which has a device of this kind in the market [34]. The function of the PM device is to insert and extract OAM

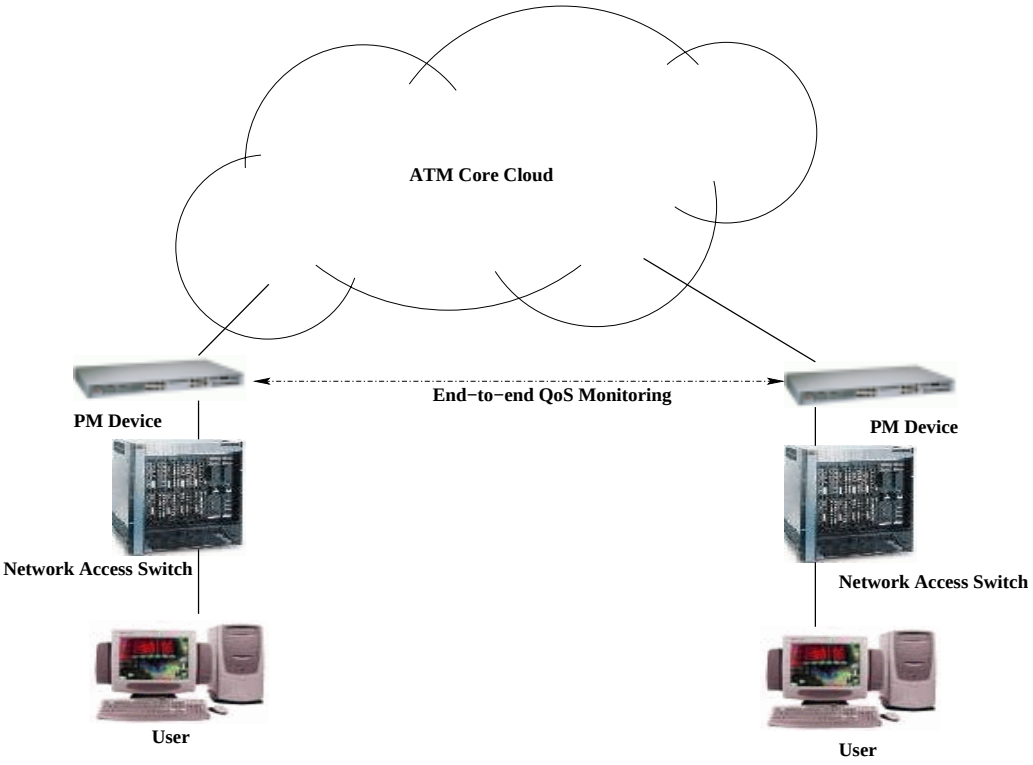

Figure 3.1: Functionality of PM devices.

performance monitoring cells, and calculate the statistics of various QoS parameters on a per connection basis. The PM device can be capable of initiating control measures as well. The functionality of the PM device can be present in the ATM access switch also.

3.2.4 Adaptation

In every switch the scheduling discipline needs to be adaptive to the end-to-end QoS measurements on real time connections. The correction in the scheduling parameters of a particular connection is in response to the end-to-end measurements performed by the PM device. The control measures can be initiated in two distinct

ways:

Centralized: The PM device taking measurements and calculating the statistics would also evaluate whether the QoS is in conformance with the guarantees, and detect any patterns of deterioration. Subsequently this device would also to identify the switch responsible for the violation or deterioration in QoS, and send a message addressed to that switch to take corrective measures and increase the priority of the connection.

Distributed: The PM device would send out the measurements or functions of the measurements (for example alarm states) in a special cell to all the switches in the path. The switches after reading the information would respond to it by changing the priority of the connection (if required) according to their resource availability.

However, methods to identify the switch causing the most deterioration in QoS do not exist. Also, in the case when the measured QoS is much better than the required, there is no mechanism for the PM device to relax the requirements at some switches. Thus the centralized scheme is inefficient and nearly unimplementable. In the distributed scheme, special OAM cells could be periodically sent out to indicate the measurement results to the switches in the path. If there is a significant difference between the measurements and the requirements, the first switch in the path - that can accommodate new scheduling parameters to alleviate the difference - takes the corrective measures. It also marks its action on the forwarded OAM cell to inform the downstream nodes. This is one simple algorithm to adapt the local schedulers at switches with respect to the end-to-end measurements. In [35], the authors propose a similar scheme for IP networks, where the

PM device changes the resource allocation of the connection in all the switches in the path.

3.2.5 Scheduling at one node

The framework poses two requirements on the schedulers of the nodes: all schedulers should be able to provide delay guarantees, and the delay requirement of the connections should be correctable (in some schedulers, if not all). This is a large class of schedulers, and hence this framework would be appropriate for networks that have different nodes running different schedulers.

The scheduler accepts the parameters and the cell arrivals of various connections as input and generates the multiplexed cell departure pattern as output. The parameters of the connection are the traffic contract and local real-time measurements like the queue depth etc. For real time connections, the traffic contract would have a local delay or jitter bound. This bound needs to be adapted to the in-service delay measurements (within the acceptable region of the CAC). The scheduler also needs to provision for the various different QoS requirements like guaranteed rate, bounded delay and bounded jitter. Such a scheduler would be introduced in the next chapter.

3.2.6 Functional components

The complete performance management architecture hence requires different functional components (figure 3.2) in the network elements according to the distributed adaptation scheme. The PM device needs to perform the following functions:

1. Monitoring - measure the QoS of a connection.

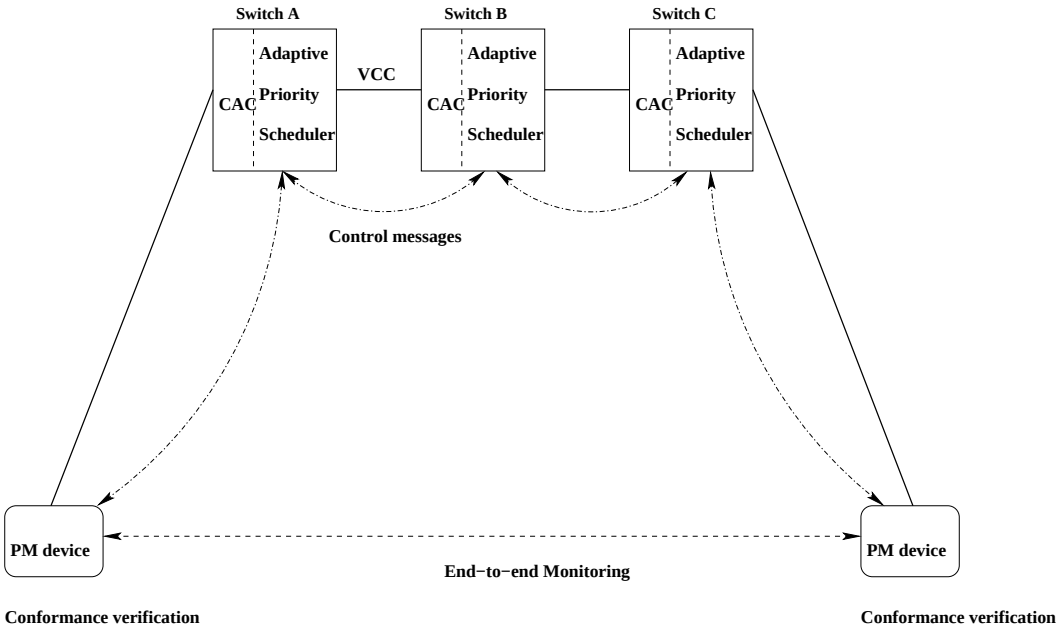

Figure 3.2: Functions of various modules.

2. Conformance verification - verify the conformance of the measurements against the guarantees at connection setup and detect any deterioration or significant changes in QoS.

3. Messaging - A mechanism to send messages (or alarms) to nodes in a connection informing the nodes of the current QoS measurements.

The switches in the path need these components:

1. Adaptive scheduling - a scheduling discipline that can adapt its parameters based on the measurements.

2. Connection Admission Control - An associated CAC algorithm that preserves

the QoS of existing connections. This is a direct function of the schedulable region of the scheduler.

3. Priority initialization and updates - an algorithm to assign priorities to connections at CAC and functions to evaluate the updates to priorities based on the incoming measurements (or alarms) information. Updates can either increase priority to improve the performance of the connection, or decrease it to improve the performance of other connections or accept new connections.

Each control action on a connection can potentially effect the QoS of other connections. However, every change in the scheduling parameters would be executed through the CAC algorithm, and therefore there should be no adverse effect on the QoS of other connections. This framework provides flexibility and control over the QoS of connections. At the same time, there exist possibilities of oscillations and instabilities, which prompt for a worst case formulated CAC algorithm and very conservative parameter change algorithm. A new delay bounding scheduler and its worst case analysis is taken up in the next chapter. The design of the priority changing algorithm at switches is a subject for future research. QoS monitoring is also explored later in the thesis.

3.3 Scope of this Work

The study of the architecture proposed in the last section involves number of questions. It is also necessary to have an implementation of the framework in a prototype ATM network in order to understand the practical issues, and to conduct some real experiments. The basic components of the framework are: scheduling at one node, end-to-end monitoring of QoS, and adaptation of schedulers in response

to the measurement results. In this work, an emphasis is made on proposing, analyzing, and implementing a new delay bounding scheduler that is a component of the proposed hierarchical scheduling discipline. Comparisons with other well known schedulers are also made. Mechanisms for in-service monitoring of delay and jitter are developed and a simple scheme for message passing between the PM device and the network nodes is also presented. The algorithm to adapt the scheduler parameters in response to the end to end measurements is a topic for further study. The implementation of the monitoring and adaptation mechanisms in the experimental switch and the development of the complete testbed for experiments are also beyond the scope of this work. The following list tabulates the contributions and and the subsequent list gives some topics for future work:

Contributions

1. Development of QoS monitoring mechanisms.
2. A novel hierarchical scheduler to guarantee the various QoS requirements.
3. An analytical study of the proposed delay bounding scheduler (HRDF) at a single node.
4. Comparison of HRDF to GPS and EDF.
5. Implementation of the scheduler in an experimental ATM switch.
6. Verification of analytical results with experiments on the switch.
7. Heuristic proposed for signaling mechanism to communicate the QoS monitoring results (needs further study).

Topics for future research

1. Algorithms to update the scheduler parameters based on the monitoring results.

2. Further study of the signaling mechanism to communicate the QoS monitoring results.

3. Implementation of monitoring and signaling mechanisms in the ATM switch.

4. Development of a testbed with reasonably large number of switches and connections to evaluate the capabilities of the performance management framework.

Chapter 4

Scheduling at One Node

A novel QoS measurements based dynamic priority scheduling algorithm is proposed in this research. This scheduling discipline is developed for the output buffered per-VC queued ATM switch architecture. The scheduling algorithms in use today are rate-based, i.e., they overlook the QoS required by a connection and assign bandwidth to the connection only on the basis of its desired rate. This couples the reserved rate of a connection to the delay it receives at every switch. Since the same average rate can be provided to a connection with widely varying cell departure patterns, the rate based scheduling schemes may not provide the required delay variation bound as well. A hierarchical priority scheduling scheme, where the priority is dynamically adjusted using end-to-end QoS measurements on a specific connection is proposed here. This scheme isolates connections requiring bandwidth guarantees and those requiring QoS performance objectives. The abstract notion of urgency is used to compare the different classes using one parameter. Urgency is defined based on the performance requirement of a connection.

4.1 Requirements for the Scheduler

The first step in developing a scheduling discipline is to understand the traffic types and buffering scheme in the switch. The premise in this research is that

the switches are output buffered and per-VC queued (Assumption 2.2.1). There are two classes of services that may require delay and loss guarantees, CBR and rt-VBR. nrt-VBR connections may require a bound on cell loss. CBR always requires a bandwidth guarantee and VBR requires bandwidth guarantees in the form of a minimum bandwidth (SCR) and high probability of switching bursts of cells (of maximum size MBS) at rates upto a maximum rate (PCR). UBR is a best effort service. It is assumed that all the traffic in the network has been policed at the access points and conforms with its UPC parameters once inside the network (Assumption 3.2.2).

For nrt-VBR streams (and CBR streams requiring no QoS guarantees), the objective is to schedule them based only on their rate requirements. Thus, in a period of time, they should receive a pre-determined share of the output link, but how the cells will be distributed in that time is not of importance. For real time streams requiring delay guarantees, the distribution of cells in a period of time is also important, even when the average share of the connection for the output link is fair. Thus these cells need to be scheduled keeping in mind tight delay requirements. If the connection is sensitive to maximum delay the cells have to be scheduled keeping in regard the time they spend in the system, rather than the rate parameters of the connection. If the stream is jitter sensitive only, the cells have to be scheduled in a manner that the pattern of arrival of cells into the switch is mimicked as closely as possible at the output link, i.e., the variance of delay in the switch has to be within tight bounds. The rate of the connection does not come into picture in this class, given that the stream is policed for rate at the network access node. If the connection is sensitive to maximum delay and delay jitter, both the requirements above need to be fulfilled. A loss sensitive connection essentially demands a certain amount of buffering in the switch, which can be

provided by either reducing the delay of the connection in the switch or providing the connection with a large buffer space.

Thus, define the following different classes of service in a switch, based on the QoS requirements:

1. Rate based and real time QoS insensitive: requiring an average bandwidth over a period of time.

2. Maximum delay sensitive: requiring bounded delay in every switch.

3. Jitter sensitive: requiring (approximately) the same departure pattern as the arrival profile of the connection. Also may require a bound on the maximum delay in a switch.

4. Loss sensitive: requiring a certain amount of buffer in the switch in addition to any of the above requirements.

5. Unspecified Bit Rate: Requiring no guarantees of any kind, a best effort service.

4.2 A Novel Hierarchical Scheduler

The proposed scheduler is hierarchical, with a higher level scheduler assigning slots to lower level schedulers for the different traffic classes (figure 4.1). Each of the per-VC queue i has an associated urgency $U_i(n)$ as a function of time slots n, defined as the urgency of the head-of-the-line cell of that queue to exit the queue at slot n. The urgency is an abstraction for quantifying relative priority of connections demanding different QoS. The calculation for urgency is based on the QoS demand of the connection and other observed parameters. The urgency of the lower-level

scheduler at a particular time is the maximum of the urgencies of the queues under it. For each slot on the output, the lower level schedulers contend and the slot is assigned to the highest urgency queue (ties are resolved randomly). Three lower level schedulers are defined:

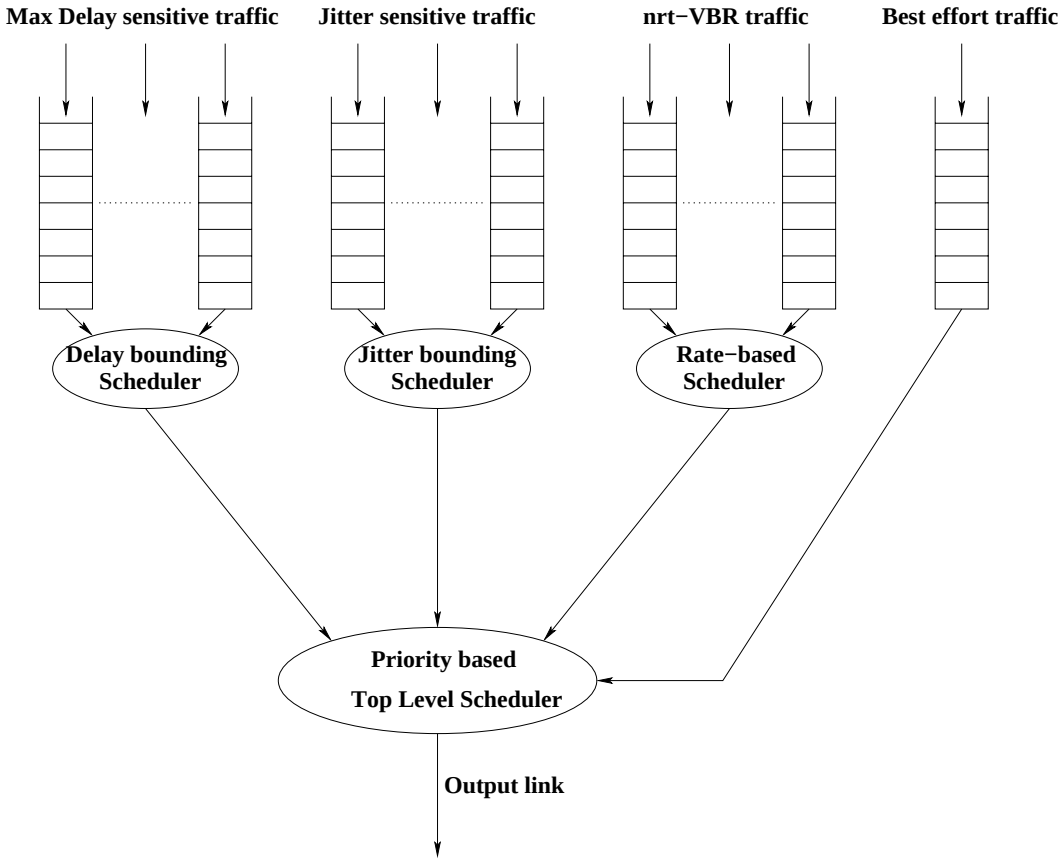

Figure 4.1: Proposed hierarchical scheduler.

Rate based: The rate based QoS insensitive traffic is switched by a round robin scheduler like WFQ. The urgency of a cell at the head of a queue (of this scheduler) to grab a slot at the output rises as the connection gets less bandwidth on the output and vice versa. A moving average of the bandwidth provided to the connection is maintained and the urgency reflects its difference with the traffic contract.

Delay based: The deadline of each cell j of queue i is the local delay bound in the switch (δ_i) added to the arrival time of the cell (A_i^j). It is calculated and stamped on the cell. The urgency of a head-of-the-line cell k, $U_i^k(n)$, to grab a particular slot on the output rises linearly with time as the deadline approaches. Thus,

$$\begin{aligned} U_i^k &= 0 \qquad n\epsilon(0, A_i^k), \\ &= \frac{(n - A_i^k)}{\delta_i} \qquad n \geq A_i^k \end{aligned} \tag{4.1}$$

The parameter δ_i^k determines the urgency - delay curve of a cell inside a switch. Henceforth, this scheduler will be referred to as Highest Relative Delay First (HRDF). Although a similar scheduler has been analyzed by Kleinrock in [36] and [37] for the case of poison arrival process and exponential departure process, the consideration of this scheduler to ATM like protocols was not found in literature. Analysis of similar dynamic priority schemes in similar scenarios is also found in Jackson's work [38] and [39].

Jitter based: The design criterion for this scheduler is that the cell should be transmitted very close to its deadline. The scheduler should pick the head-of-the-line cell whose deadline is near the current time. Thus the urgency of a cell at the head of the queue should rise to maximum very sharply at the time of the deadline. Thus,

$$\begin{aligned} U_i^k &= 0 \qquad n\epsilon(0, A_i^k + \beta_i), \\ &= \frac{(n - A_i^k)}{\delta_i} \qquad n \geq A_i^k + \beta_i \end{aligned} \tag{4.2}$$

The parameter β_i provides margin to smooth local fluctuations in the delay and make sure that most of the cells reach the head-of-the-line before it. It holds the cells in the switch for a time (should be more than the average delay time for a similar delay queue) so that the probability of going out is much more later. The parameter (δ_i - β_i) provides a rough bound on the delay variation expected in one switch. In this work, the jitter scheduler is merged with the HRDF scheduler by forcing $\beta_i = 0$. This makes the analysis more tractable and the implementation easier. This would also be the case where the jitter sensitive connection also expects a very tight delay bound.

If the slot is not needed by any of the lower level schedulers, it is assigned to the best effort traffic queue. The parameters β_i and δ_i have to be chosen so as to provide a fair distribution of resources amongst the various types of connections. Note an important attribute: the concept of fairness here is very different from that of providing proportionate residual bandwidth as in the guaranteed bandwidth schemes. Fairness now extends to comparing the resources used by connections demanding different QoS requirements.

These parameters are also adapted according to network measurements. For example, if a connection is incurring a high average delay, the parameter δ_i can either be decreased at the bottleneck node or at all the nodes in its path. Thus, the assignment of switch parameters at connection startup need not be exact as they can be tuned with time. If a violation with respect to loss parameters is detected, the switches can either increase the buffer allocated to the connection or increase its priority in order to reduce the buffering inside the network. Monitoring becomes very useful as the inaccurate initialization of priority at the connection setup can be corrected if required. In this way, end-to-end bounds on the quality of service can be provided to connections, without the prior knowledge of the number

of hops in the circuit.

4.3 Assumptions and Definitions

The analysis of the HRDF scheduler is performed under the traffic flow constraints of ATM in the next section. This section presents some relevant definitions and assumptions.

The flow control mechanism in ATM is called Usage Parameter Control (UPC). It constrains a flow to its PCR, SCR and MBS specifications using two leaky buckets. As mentioned earlier, it is assumed that flows are policed using this algorithm at the ATM network access point (Assumption 3.2.2). Thus any traffic entering the network is assumed to be constrained by UPC dual leaky bucket constraint.

In the terminology of [11], for a connection $i\epsilon\{1, ..., N\}$ the SCR corresponds to the average rate ρ_i, MBS to the size of the bucket σ_i, and PCR to the C_i of the session. The constraint imposed by the leaky bucket is as follows: If $A_i(\tau, t)$ is the amount of flow leaving the bucket and entering the network in time (τ, t) then

$$A_i(\tau, t) \leq min\{(t-\tau)C_i, \sigma_i + \rho_i(t-\tau)\}, \quad \forall t \geq \tau \geq 0 \tag{4.3}$$

Consider the session i conforming to (σ_i, ρ_i, C_i) (or $A_i \sim (\sigma_i, \rho_i, C_i)$). Let the traffic model be fluid. This means that the traffic can arrive and be serviced in infinitesimal units. A session i is defined to be greedy at time τ if it uses as many tokens as possible for all times $t \geq \tau$:

$$A_i^{\tau}(\tau, t) = min\{(t-\tau)C_i, l_i(\tau) + \rho_i(t-\tau)\} \quad \forall t \geq \tau \tag{4.4}$$

where $l_i(\tau)$ is the number of tokens in the bucket of session i at time τ. Thus $l_i(t) \leq \sigma_i$, and if the connection i starts with a full bucket at time 0, $l_i(0) = \sigma_i$.

Define the aggregate output link capacity to be C. Then the CAC guarantees that $C \geq \sum_{i=1}^{n} \rho_i$. For simplicity of analysis, assume $C_i = \infty$ to start with. The arrival constraint is thus modified to

$$A_i(\tau, t) \leq \sigma_i + \rho_i(t - \tau), \quad \forall t \geq \tau \geq 0 \tag{4.5}$$

for all sessions. It is assumed that the arrival function is left continuous. When the session i is greedy from time τ, Eqn. (4.4) reduces to

$$A_i^{\tau}(\tau, t) = l_i(\tau) + \rho_i(t - \tau). \tag{4.6}$$

Note that if the session is greedy after time τ, $l_i(t) = 0$ for all $t \geq \tau$.

Let $S_i(\tau, t)$ be the amount of session i traffic served in the interval $(\tau, t]$. Thus $S_i(0, t)$ is continuous and non-decreasing in t. The session i backlog at time τ is defined as (figure 4.2):

$$Q_i(\tau) = A_i(0, \tau) - S_i(0, \tau). \tag{4.7}$$

The delay of session i time τ is defined as the time that traffic arriving at time τ spends in the queue (figure 4.2):

$$D_i(\tau) = inf\{t \geq \tau : S_i(0, t) = A_i(0, \tau)\} - \tau. \tag{4.8}$$

The maximum delay for a connection, over all the arrival functions constrained by Eqn. (4.3), is:

$$D_i^* = \max_{A_1, \ldots, A_N} \max_{\tau \geq 0} D_i(\tau). \tag{4.9}$$

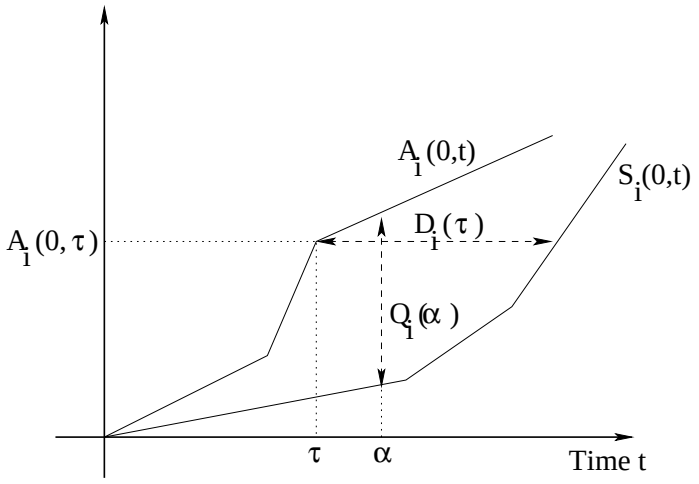

Figure 4.2: Example of $D_i(\tau)$ and $Q_i(\alpha)$.

Define a system busy period to be a maximal interval B such that for any $\tau, t \, \epsilon B, \tau \leq t$:

$$\sum_{i=1}^{N} S_i(\tau, t) = t - \tau.$$

Since the system is work conserving, if $B = [t_1, t_2]$, then $\sum_{i=1}^{N} Q_i(t_1) = \sum_{i=1}^{N} Q_i(t_2) = 0$.

HRDF

Consider the HRDF scheduler with N connections, each characterized by A_i ~ (σ_i, ρ_i, C_i) and requiring a local delay bound of δ_i where $i = \{1 \ldots N\}$. Define $x_i = \frac{1}{\delta_i}$. The urgency of a cell is defined as $U_i(\tau) = D_i(\tau) * x_i$. Define U_i^* as the maximum urgency of a connection over time, over all arrival patterns constrained by A_is:

$$U_i^* = \max_{A_1, \ldots, A_N} \max_{\tau \geq 0} U_i(\tau). \tag{4.10}$$

Define U^* as the maximum over the maximum urgencies of all connections:

$$U^* = \max_{i=1,..,N} U_i^*. \tag{4.11}$$

The schedulability condition for a given set of connection parameters is therefore:

$$\boxed{U^* \leq 1.} \tag{4.12}$$

as it ensures that there is no delay constraint violation in any case.

4.4 Fluid Analysis of HRDF

Lemma 4.4.1 The following is a practical constraint:

$$\frac{\sigma_i x_i}{\rho_i} \geq 1 \quad \forall i \tag{4.13}$$

Proof: For a rate proportional scheduler, the guaranteed rate to connection i would be ρ_i. Thus the maximum delay would be $\frac{\sigma_i}{\rho_i}$ for a greedy connection. The use of a delay guaranteeing scheduler is justified only if the required delay bound is less than this, i.e., $\delta_i \leq \frac{\sigma_i}{\rho_i}$. Thus, $\frac{\sigma_i x_i}{\rho_i} \geq 1$. □

Lemma 4.4.2 Simultaneous Arrivals: Consider a delay scheduler with N connections ordered such that $x_1 \geq .... \geq x_N$. If all the connections come greedy at time 0, the various peaks in the urgency curve are:

$$U_j^S = \frac{\sum_{i \leq j} \sigma_i x_j}{C - \sum_{i \leq j} \rho_i (1 - \frac{x_j}{x_i})}.$$

Proof: An extension of case 1 analysis in lemma A.1.1 with urgency curve as shown in figure 4.3. □

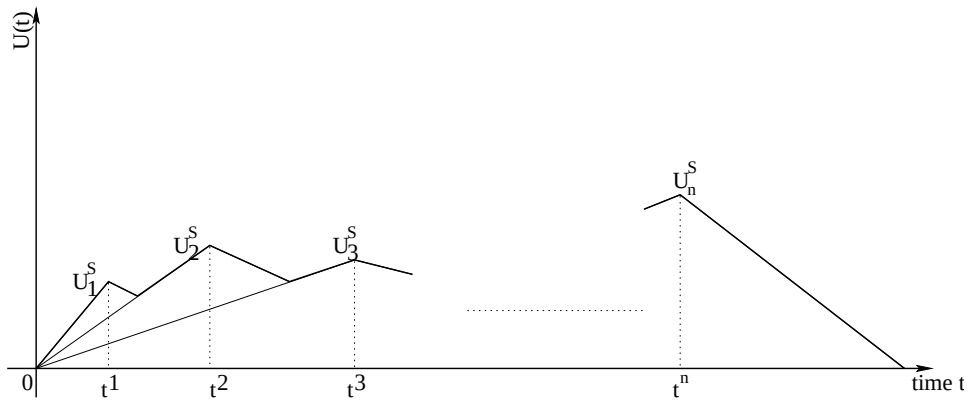

Figure 4.3: Analysis for all connections greedy at time 0.

Lemma 4.4.3 Cascading Arrivals: Consider a delay scheduler with N connections ordered such that $x_1 \geq .... \geq x_N$. If the connection N arrives greedy at time 0, connection $N-1$ arrives at a time such that its urgency curve hits that of connection N at the peak, and so on, then the maximum urgency reached is:

$$\sum_i \frac{\sigma_i x_i}{C}.$$

Proof: An extension of case 2 analysis in lemma A.1.1 with urgency curve as shown in figure 4.4. □

Lemma 4.4.4 Consider a set of connections Θ. Given that the constraint in lemma 4.4.1 is satisfied, and the connections are schedulable if they arrive in a cascading manner, the urgency of the cascading connections is larger than all the peaks in the urgency curve obtained when all connection arrive simultaneously.

Proof: Let the connections in Θ be ordered $1, ..., k$ in increasing order of x_i. Consider the last peak in the case of simultaneous arrivals

$$U_k^S = \frac{\sum_{i \leq k} \sigma_i x_k}{C - \sum_{i \leq k} \rho_i (1 - \frac{x_k}{x_i})}.$$

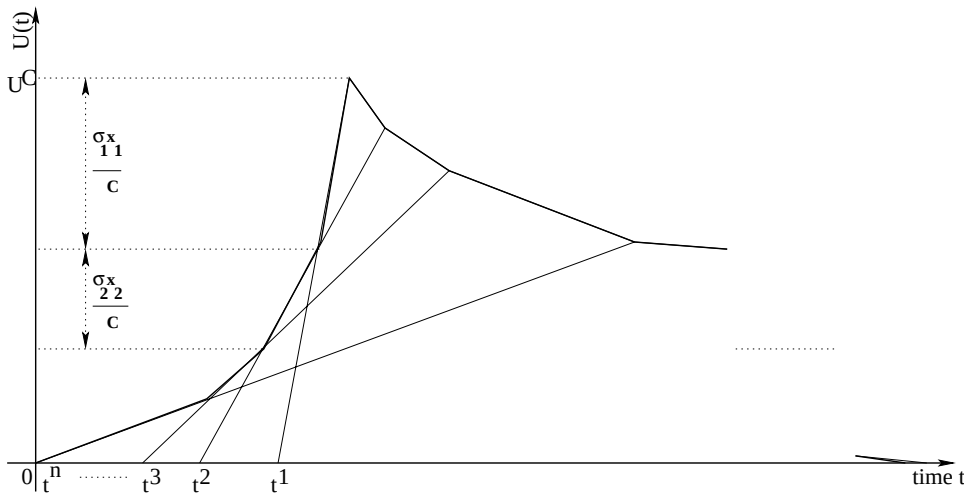

Figure 4.4: Analysis for all connections arriving in a cascading pattern.

The highest peak of the cascading arrivals is

$$U^C = \sum_{i=1}^{k} \frac{\sigma_i x_i}{C}.$$

To prove $U_k^S \leq U^C$:

$$\begin{aligned}
U_k^S &\leq U^C \\
or, \quad \frac{\sum_{i\leq k} \sigma_i x_k}{C - \sum_{i\leq k} \rho_i (1 - \frac{x_k}{x_i})} &\leq \sum_{i=1}^{k} \frac{\sigma_i x_i}{C} \\
or, \quad \sum_{i\leq k} \sigma_i x_k C &\leq (\sum_{i=1}^{k} \sigma_i x_i)(C - \sum_{j\leq k} \rho_j (1 - \frac{x_k}{x_j}) \\
or, \quad \sigma_k x_k (\sum_{i\leq k} \rho_i (1 - \frac{x_k}{x_i})) &\leq \sum_{i=1}^{k-1} \sigma_i x_i \{C(1 - \frac{x_k}{x_i}) - \sum_{j\leq k} \rho_j (1 - \frac{x_k}{x_j})\} \\
&\leq \sum_{i=1}^{k-1} \sigma_i x_i \{(C - \rho_i)(1 - \frac{x_k}{x_i})\} \\
&\quad - \sum_{i=1}^{k-1} \sigma_i x_i \{ \sum_{j\leq k, j\neq i} \rho_j (1 - \frac{x_k}{x_j})\} \\
&\leq \sum_{i=1}^{k-1} \sigma_i x_i \{(C - \rho_i)(1 - \frac{x_k}{x_i})\}
\end{aligned}$$

$$
\begin{aligned}
& & & & & -\sum_{i=1}^{k}\rho_i(1-\frac{x_k}{x_i})\{\sum_{j\le k-1, j\neq i}\sigma_j x_j\} \\
or,\ & \sum_{i=1}^{k}\rho_i(1-\frac{x_k}{x_i})\{\sum_{j\le k, j\neq i}\sigma_j x_j\} & \le & \sum_{i=1}^{k-1}\sigma_i x_i\{(C-\rho_i)(1-\frac{x_k}{x_i})\} \\
or,\ & \sum_{i=1}^{k-1}\rho_i(1-\frac{x_k}{x_i})\{\sum_{j\le k}\sigma_j x_j\} & \le & \sum_{i=1}^{k-1}\sigma_i x_i\{C(1-\frac{x_k}{x_i})\} \\
or,\ & \sum_{i=1}^{k-1}\rho_i(1-\frac{x_k}{x_i})\{\sum_{j\le k}\frac{\sigma_j x_j}{C}\} & \le & \sum_{i=1}^{k-1}\sigma_i x_i(1-\frac{x_k}{x_i})
\end{aligned} \tag{4.14}
$$

From lemma 4.4.1, it is given that

$$\frac{\sigma_i x_i}{\rho_i} \ge 1 \quad \forall i.$$

and that the cascading arrivals case is schedulable

$$\sum_{i=1}^{k}\frac{\sigma_i x_i}{C} \le 1.$$

Therefore,

$$
\begin{aligned}
& \sum_{i=1}^{k}\frac{\sigma_i x_i}{C} & \le & \frac{\sigma_i x_i}{\rho_i} \quad \forall i \\
or, \quad & \rho_i(1-\frac{x_k}{x_i})\sum_{i=1}^{k}\frac{\sigma_i x_i}{C} & \le & \sigma_i x_i(1-\frac{x_k}{x_i}) \quad \forall i \\
or, \quad & \sum_{i=1}^{k-1}\rho_i(1-\frac{x_k}{x_i})\{\sum_{j\le k}\frac{\sigma_j x_j}{C}\} & \le & \sum_{i=1}^{k-1}\sigma_i x_i(1-\frac{x_k}{x_i})
\end{aligned} \tag{4.15}
$$

This proves that $U_k^S \le U_k^C$ under the given conditions.

Consider $j < k$, and its peak in the simultaneous arrival case U_j^S. By the above analysis, $U_j^S \le U_j^C$. Now $U_j^C \le U_k^C$, $=> U_j^S \le U_k^C$. Therefore, the urgency of the cascading arrivals case is larger than all the peaks of the simultaneous arrival urgency curve. □

Lemma 4.4.5 Cascade + Simultaneous Arrivals: Consider a delay scheduler with N connections ordered such that $x_1 \geq .... \geq x_N$. If connections in set θ arrive in a cascading manner starting at time 0 and the remaining connections arrive simultaneously at time t_S such that the urgency of connection with lowest index in θ^C intersects the system urgency at U_θ^C, then the highest system urgency is not more than $U_{1,...,N}^C$.

Proof: The various scenarios that can happen in this pattern of traffic arrivals can be explored by considering a smaller set of connections. Suppose that there are three connections and the set θ is composed of only one connection. This can be either connection 2 or connection 3.

Consider $\theta = \{2\}$ first depicted in figure 4.5. Traffic from connection 2 arrives at time 0 and traffic from connections 1 and 3 arrives at the same time, $t_0 = \frac{\sigma_2}{C}(1 - \frac{x_2}{x_1})$. Connection 1 starts to get service at $t_1 = \frac{\sigma_2}{C}$, and its burst is serviced till time $t_2 = \frac{\sigma_1+\sigma_2}{C}$, after which the urgency of 1 starts to decrement. At time t_3, the urgency of 1 and 2 become equal. Then,

$$\begin{aligned} [(t_3 - t_0) - (t_3 - t_2)\frac{C}{\rho_1}]x_1 &= t_3 x_2 \\ or,\ \ t_3 = \frac{\sigma_2}{C} + \frac{\sigma_1}{C - \rho_1(1 - \frac{x_2}{x_1})} & \end{aligned} \tag{4.16}$$

Then, these connections would be served in the ratio $(r_1, r_2 = (1 - r_1))$ at $t\epsilon(t_3, t_4)$:

$$\begin{aligned} [t - \frac{(t - t_3)Cr_2}{\rho_2}]x_2 &= [t - t_3(1 - \frac{x_2}{x_1}) - \frac{(t - t_3)Cr_1}{\rho_1}]x_1 \\ \frac{x_2}{x_1} - \frac{C(1 - r_1)}{\rho_2}\frac{x_2}{x_1} &= 1 - \frac{Cr_1}{\rho_1} \\ r_1 &= \frac{\frac{x_2}{x_1} + \frac{\rho_2}{C}(1 - \frac{x_2}{x_1})}{\frac{x_2}{x_1} + \frac{\rho_2}{\rho_1}} \end{aligned} \tag{4.17}$$

$$r_2 = \frac{\frac{\rho_2}{\rho_1} - \frac{\rho_2}{C}(1 - \frac{x_2}{x_1})}{\frac{x_2}{x_1} + \frac{\rho_2}{\rho_1}} \tag{4.18}$$

To calculate t_4:

$$\begin{aligned}
[t_4 - \frac{(t_4 - t_3)Cr_2}{\rho_2}]x_2 &= (t_4 - t_0)x_3 \\
or,\ \ t_4[\frac{\frac{C}{\rho_1} - (1 - \frac{x_2}{x_1})}{\frac{x_2}{x_1} + \frac{\rho_2}{\rho_1}} + \frac{x_3}{x_2} - 1] &= \frac{\sigma_2(C - \rho_1(1 - \frac{x_2}{x_1})) + \sigma_1 C}{C(\rho_2 + \rho_1\frac{x_2}{x_1})} + \frac{\sigma_2}{C}\frac{x_3}{x_2}(1 - \frac{x_2}{x_1}) \\
or,\ \ t_4 &= \frac{\sigma_2}{C} + \frac{\frac{\sigma_2}{C}(\rho_2 + \rho_1\frac{x_2}{x_1})x_2(1 - \frac{x_3}{x_1}) + \sigma_1 x_2}{(C - \rho_1 - \rho_2)x_2 + (\rho_2 + \rho_1\frac{x_2}{x_1})x_3} \\
or,\ \ t_4 &= \frac{\sigma_2}{C} + \frac{\sigma_2(\rho_2 + \rho_1\frac{x_2}{x_1})(1 - \frac{x_3}{x_1}) + \sigma_1}{C - \rho_1(1 - \frac{x_3}{x_1}) - \rho_2(1 - \frac{x_3}{x_2})}
\end{aligned} \tag{4.19}$$

At $t\epsilon(t_4, t_5)$, the connections are serviced in the ratio $(s_1, s_2, s_3 = (1 - s_1 - s_2))$:

$$\begin{aligned}
[t_4 - \frac{(t_4 - t_3)Cr_2}{\rho_2} + (t - t_4)(1 - \frac{Cs_2}{\rho_2})]x_2 &= [t_3\frac{x_2}{x_1} + (t_4 - t_3)(1 - \frac{Cr_1}{\rho_1}) \\
&\quad +(t - t_4)(1 - \frac{Cs_1}{\rho_1})]x_1 \\
or,\ \ [(t - t_4)(1 - \frac{Cs_2}{\rho_2})]x_2 &= [(t - t_4)(1 - \frac{Cs_1}{\rho_1})]x_1 \\
+[t_4 - \frac{(t_4 - t_3)Cr_2}{\rho_2}]x_2 &\quad +[t_4 - t_3(1 - \frac{x_2}{x_1}) - \frac{(t_4 - t_3)Cr_1}{\rho_1}]x_1
\end{aligned}$$

The second terms of both equations are equal by the definition of t_4. So,

$$(1 - \frac{Cs_2}{\rho_2})x_2 = (1 - \frac{Cs_1}{\rho_1})x_1.$$

A similar equation can also be written for 3 and 1:

$$\begin{aligned}
[t_4 - \frac{(t_4 - t_3)Cr_2}{\rho_2} + (t - t_4)(1 - \frac{Cs_2}{\rho_2})]x_2 &= (t - t_0)x_3 \\
or,\ \ [(t - t_4)(1 - \frac{Cs_2}{\rho_2})]x_2 &= (t - t_4)x_3 \\
+[t_4 - \frac{(t_4 - t_3)Cr_2}{\rho_2}]x_2 &\quad +(t_4 - t_0)x_3
\end{aligned}$$

Again, the second terms of both equations are equal by the definition of t_4. So,

$$(1 - \frac{Cs_2}{\rho_2})x_2 = x_3.$$

Thus,

$$\begin{aligned} s_1 &= \frac{\rho_1}{C}(1 - \frac{x_3}{x_1}) \\ s_2 &= \frac{\rho_2}{C}(1 - \frac{x_3}{x_2}) \\ s_3 &= 1 - \frac{\rho_1}{C}(1 - \frac{x_3}{x_1}) - \frac{\rho_2}{C}(1 - \frac{x_3}{x_2}) \end{aligned} \tag{4.20}$$

At t_5,

$$\begin{aligned} (t_5 - t_4)s_3C &= \sigma_3 \\ or \;\; t_5 &= \frac{\sigma_2}{C} + \frac{\sigma_2(\frac{\rho_2}{C} + \frac{\rho_1}{C}\frac{x_2}{x_1})(1 - \frac{x_3}{x_1}) + \sigma_1 + \sigma_3}{C - \rho_1(1 - \frac{x_3}{x_1}) - \rho_2(1 - \frac{x_3}{x_2})} \end{aligned} \tag{4.21}$$

$$\tag{4.22}$$

Therefore,

$$\begin{aligned} U_5 &= (t_5 - t_0)x_3 \\ &= \left\{ \frac{\sigma_1 + \sigma_2\{1 - (1 - \frac{\rho_2}{C})(1 - \frac{x_2}{x_1})\} + \sigma_3}{C - \rho_1(1 - \frac{x_3}{x_1}) - \rho_2(1 - \frac{x_3}{x_2})} \right\} x_3 \\ &\leq U^S_{\{1,2,3\}} \\ &\leq U^C_{\{1,2,3\}} \end{aligned} \tag{4.23}$$

After t_5, the urgency of the system strictly decreases. Thus it is proved that in this case the urgency of the system never goes above the simultaneous arrival urgency.

Now consider $\theta = \{3\}$. There are two possibilities here after time t_2. The urgency of 2 can either overtake the urgency of 3 or not depending on the parameters. Consider first that it does not overtake. In this case the analysis is very similar to the first case:

$$
\begin{aligned}
t_0 &= \frac{\sigma_3}{C}(1-\frac{x_3}{x_1}) \\
t_1 &= \frac{\sigma_3}{C} \\
t_2 &= \frac{\sigma_3+\sigma_1}{C} \\
t_3 &= \frac{\sigma_3}{C} + \frac{\sigma_1}{C-\rho_1(1-\frac{x_3}{x_1})} \\
r_1 &= \frac{\frac{x_3}{x_1}+\frac{\rho_3}{C}(1-\frac{x_3}{x_1})}{\frac{x_3}{x_1}+\frac{\rho_3}{\rho_1}} \\
r_2 &= \frac{\frac{\rho_3}{\rho_1}-\frac{\rho_3}{C}(1-\frac{x_3}{x_1})}{\frac{x_3}{x_1}+\frac{\rho_3}{\rho_1}} \\
t_4 &= \frac{\sigma_3}{C} + \frac{\sigma_3(\frac{\rho_3}{C}+\frac{\rho_1}{C}\frac{x_3}{x_1})(1-\frac{x_2}{x_1})+\sigma_1}{C-\rho_1(1-\frac{x_2}{x_1})-\rho_3(1-\frac{x_2}{x_3})} \\
s_1 &= \frac{\rho_1}{C}(1-\frac{x_2}{x_1}) \\
s_3 &= \frac{\rho_3}{C}(1-\frac{x_3}{x_1}) \\
s_2 &= 1-\frac{\rho_1}{C}(1-\frac{x_2}{x_1})-\frac{\rho_3}{C}(1-\frac{x_3}{x_1}) \\
t_5 &= \frac{\sigma_3}{C} + \frac{\sigma_3(\frac{\rho_3}{C}+\frac{\rho_1}{C}\frac{x_3}{x_1})(1-\frac{x_2}{x_1})+\sigma_2+\sigma_1}{C-\rho_1(1-\frac{x_2}{x_1})-\rho_3(1-\frac{x_2}{x_3})} \\
U_5 &= \left\{\frac{\sigma_1+\sigma_2+\sigma_3\{1-(1-\frac{\rho_3}{C})(1-\frac{x_3}{x_1})\}}{C-\rho_1(1-\frac{x_2}{x_1})-\rho_3(1-\frac{x_2}{x_3})}\right\}x_2 \\
&\leq U^C_{\{1,2,3\}}
\end{aligned}
$$

It can be shown that $U_5 \leq U^C_{\{1,2,3\}}$ when $U^C_{\{1,2,3\}} \leq 1$ and $\sigma_i x_i \geq \rho_i \ \forall i$ (lemma 4.4.1).

The second possibility is that the urgency of connection 2 meets the system urgency before that of connection 3 meets. In this case, as shown in figure 4.6,

the system urgency in the interval (t_0, t_5) behaves equivalent to the case when connections 1 and 2 arrive at the same time, with $\sigma_1^{eff} = \sigma_1 + \sigma_3 \frac{x_3}{x_1}$. It is easy to show that the peaks here are always less than the peaks attained in the case of simultaneous arrival of all the three connections.

Hence it is shown that the system urgency is always lower than the urgency of the system under the case of simultaneous arrivals for the case of three connections. If the set θ is composed of more connections, the analysis is very much the same as the peaks are still obtained in the manner described above. If the set θ^C is composed of more than two connections, the behavior would be similar to the two cases above and the peaks would be still bounded. Thus in the general case also, the profiles of the urgency curve would exhibit similar characteristics. From the above analysis it is clear that the urgency in no case would be more the case of simultaneous arrivals. □

Theorem 4.4.6 Consider a HRDF scheduler with N connections ordered such that $x_1 \geq .... \geq x_N$. Then under the practical constraint of lemma 4.4.1 the schedulability region is defined by the equation :

$$\boxed{\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} \leq 1.} \tag{4.24}$$

Proof: The cascading arrival pattern achieves the urgency on the LHS of Eqn. (4.24). Thus, if $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} > 1$, the connections are not schedulable because there exists an arrival pattern which will violate the delay requirement of at least one connection.

Consider the case when $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} \leq 1$ under the practical constraint of lemma 4.4.1. As is observable from the analysis in the Appendix, during the rising edges of the urgency, the intersection points should be at the corners points of the curve.

If the intersection is at the upper corner, the arrivals are cascading and if the intersection is at the lower corner, the arrivals are simultaneous. Thus the N connections can be combined together into groups that either arrive simultaneously or in a cascading manner. The relative times of arrivals of these groups are such that the urgency is maximized. Assume that the busy period starts at time 0 and the subset of connections θ arrives at time 0. If θ has two connections, then lemma A.1.1 applied under 4.4.1 says that the maximum urgency achieved would be in the case of cascading arrivals.

For any set θ, from lemma 4.4.4 it is known that the achieved urgency would be higher if the connections actually arrived in cascade pattern starting at time 0. Assume that the maximum urgency of θ is realized for cascading arrivals. Now suppose that another subset of connection S arrives at time greater than 0 such that they intersect the highest point of the urgency curve of the subset θ. If these connections arrive in a cascade manner, it is equivalent to the subset $\theta \bigcup S$ arriving in a cascade manner. Otherwise if the connections arrive simultaneously, the maximum urgency would be bounded by the maximum urgency generated by the subset $\theta \bigcup S$ arriving in a cascade manner by lemma 4.4.5. This argument can then be extended until all connections are exhausted. Given the initial condition of two connections, this inductive argument proves that the maximum system urgency for these connections would then be bounded by $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C}$. Thus if $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} \leq 1$, the connections are always schedulable. □

Lemma 4.4.7 The schedulability region of HRDF when C_i is finite is at least as much when $C_i = \infty$.

Proof: The worst case traffic arrival for traffic conforming with the leaky bucket constraint (σ_i, ρ_i, C_i) is no more than the worst case arrival for $(\sigma_i, \rho_i, \infty)$. Thus

the above analysis holds for finite C_i and the same schedulability region applies here also. □

4.5 Comparison of schedulability regions of HRDF and GPS

As observed in the appendix, calculation of maximum delays for individual connections is a difficult task for both HRDF and GPS. Thus for the case of general N, comparison will be made between the CAC of the two policies. In practice also, the schedulability region is a more important concept than the delay statistics of a set of schedulable connections.

Theorem 4.5.1 A set of connections satisfying the following conditions is schedulable under GPS:

$$\sigma_i x_i \geq \rho_i \quad \forall i, \tag{4.25}$$

$$\sum_{i=1}^{N} \sigma_i x_i \leq C \tag{4.26}$$

and,

$$\sum_{i=1}^{N} \rho_i \leq C. \tag{4.27}$$

Proof: As explained by Parekh and Gallagher in [11], the necessity of Eqn. (4.27) is apparent. For Eqn. (4.26), consider the following: Let ϕ_i be the weight of connection i. From [11], the worst case delays for all connections are attained when all connections arrive greedy at the same time ($t = 0$). At the start of the

busy period, when the queues of all the connections are non-zero (this time would be finite, given that $\sigma_i > 0 \quad \forall i$) the bandwidth assignment to the connections is

$$g_i = \frac{\phi_i}{\sum_{i=0}^{N} \phi_i} C \tag{4.28}$$

It is also true that the instantaneous rate provided to the connection i at time $t > 0$ is always greater than or equal to g_i, i.e., $r_i(t) \geq g_i \quad \forall t > 0$. Let the maximum delay of a connection under GPS be G_i^*. Then

$$G_i^* \leq \frac{\sigma_i}{g_i} \tag{4.29}$$

if $g_i \geq \rho_i$. The delay bound required for each connection is δ_i. Consider the connection $L(1)$ which ends its busy period first. Then $G_{L(1)}^* = \frac{\sigma_{L(1)}}{g_{L(1)}}$ and also $G_i^* < \frac{\sigma_i}{g_i} \quad \forall i \neq L(1)$. Choose $\phi_i = \sigma_i x_i$. Since $g_i \geq \rho_i$ under Eqn. (4.26),

$$U_{L(1)}^* = G_{L(1)}^* x_{L(1)} = \frac{\sum_{i=1}^{N} \sigma_i x_i}{C}$$

and,

$$U_i^* = G_i^* x_i \leq \frac{\sum_{i=1}^{N} \sigma_i x_i}{C} \quad \forall i \neq L(1)$$

Therefore,

$$U_{GPS}^* = \max_{i=1,\ldots,N} U_i^* = U_{L(1)}^* = \frac{\sum_{i=1}^{N} \sigma_i x_i}{C}$$

Thus under the Eqn. (4.26), $U_{GPS}^* \leq 1$ and $\sum_{i=1}^{N} g_i \leq C$. It can be concluded that the connections are schedulable under the given conditions. □

Theorem 4.5.2 The schedulable region of GPS is at least as much as that of HRDF under the practical constraint of lemma 4.4.1.

Proof: From theorem 4.4.6 The schedulable region of HRDF is characterized by $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} \leq 1$. The lemma 4.5.1 says that connections in this region are necessarily schedulable in GPS. It can be easily shown that some connections, not in that region, are also schedulable by GPS. However the characterization of the complete schedulability region of GPS is very difficult and was also not found in literature. The choice of the weights for individual connections in order to satisfy the delay constraints is also much more difficult in the region $\sum_{i=1}^{N} \frac{\sigma_i x_i}{C} > 1$ compared to the choice $\phi_i = \sigma_i x_i$ for the complement region. Thus the schedulability region of GPS is at least as large as that of HRDF, however the schedulability region of HRDF is characterizable and is equivalent to the easily utilizable region for GPS. □

4.6 Comparison of HRDF with EDF

Lemma 4.6.1 The schedulability region of the EDF scheduler for a set of N connections ordered such that $x_1 \geq ... \geq x_N$ is described by the equations:

$$\frac{\sum_{i=1}^{k} \sigma_i x_k}{C - \sum_{i=1}^{k-1} \rho_i (1 - \frac{x_k}{x_i})} \leq 1 \quad \forall\, k \,\epsilon\, \{1, ..., N\}$$

Proof: Equations 23, 24 and 25 in [17] give us the schedulability region of the EDF scheduler. To be consistent with assumptions in this thesis, let $L_{max} \to 0$. Those equations can be then written as:

$$\begin{aligned} 0 &\leq D_1 \\ \sum_{i=1}^{N} (\sigma_i + \rho_i (t - D_i)) U(t - D_i) &\leq Ct \quad for\ 0 \leq t \leq D_N \\ \sum_{i=1}^{N} (\sigma_i + \rho_i (t - D_i)) &\leq Ct \quad for\ t \geq D_N \end{aligned} \tag{4.30}$$

These can be written as:

$$
\begin{aligned}
0 &\leq Ct \quad for\ 0 \leq t \leq D_1 \\
\frac{\sigma_1 - \rho_1 D_1}{C - \rho_1} &\leq t \quad for\ D_1 \leq t \leq D_2 \\
\frac{\sum_{i=1}^{j} \sigma_i - \sum_{i=1}^{j} \rho_i D_i}{C - \sum_{i=1}^{j} \rho_i} &\leq t \quad for\ D_j \leq t \leq D_{j+1} \\
&\vdots \\
\frac{\sum_{i=1}^{N} \sigma_i - \sum_{i=1}^{N} \rho_i D_i}{C - \sum_{i=1}^{N} \rho_i} &\leq t \quad for\ D_N \leq t
\end{aligned} \tag{4.31}
$$

Evaluating these equations at the lower boundary is necessary and sufficient, and since $0 \leq Ct \quad for\ 0 \leq t \leq D_1$: is always true:

$$
\frac{\sum_{i=1}^{j} \sigma_i x_j}{C - \sum_{i=1}^{j} \rho_i (1 - \frac{x_j}{x_i})} \leq 1 \quad \forall\ j\ \epsilon \{1, ..., N\}
$$

□

This set of equations is achieved when all connections arrive in a greedy manner simultaneously at time 0. It is notable that this is also the set of equations obtained in the case of simultaneous greedy arrivals to a HRDF scheduler (lemma 4.4.2). Although HRDF behaves in the same manner as EDF in this case, the urgency of the cascading arrivals is much higher in HRDF than in EDF.

Let U_i^S be maximised at $i = k$. Consider the difference $U_k^C - U_k^S$:

$$
\begin{aligned}
U_k^C - U_k^S &= \frac{\sum_{i=1}^{k} \sigma_i x_i}{C} - \frac{\sum_{i=1}^{k} \sigma_i x_k}{C - \sum_{i=1}^{k} \rho_i (1 - \frac{x_j}{x_i})} \\
&= \frac{\sum_{i=1}^{k} \{\sigma_i x_i - \rho_i \left[\frac{\sum_{i=1}^{k} \sigma_i x_i}{C} \right]\} (1 - \frac{x_j}{x_i})}{C - \sum_{i=1}^{k} \rho_i (1 - \frac{x_j}{x_i})}
\end{aligned} \tag{4.32}
$$

Assume that the connections are schedulable using HRDF. Thus, $\frac{\sum_{i=1}^{k} \sigma_i x_i}{C} \leq 1$. Also from the assumption of lemma 4.4.1, $\sigma_i x_i \geq \rho_i$. Assume that $\sigma_i x_i \gg \rho_i$ for

the sake of simplicity. Therefore,

$$\begin{aligned} U_k^C - U_k^S &\approx \frac{\sum_{i=1}^{k} \sigma_i (x_i - x_k)}{C - \sum_{i=1}^{k} \rho_i (1 - \frac{x_i}{x_i})} \\ &= \left[\frac{\sum_{i=1}^{k} \sigma_i x_i}{\sum_{i=1}^{k} \sigma_i x_k} - 1 \right] U_k^S \end{aligned}$$

Or,

$$\frac{U_k^S}{U_k^S} \approx \frac{\sum_{i=1}^{k} \sigma_i \frac{x_i}{x_k}}{\sum_{i=1}^{k} \sigma_i} \tag{4.33}$$

Although U_k^S is the highest urgency figure for the EDF scheduler, the region for HRDF is really U_N^C which will be more than U_k^C. Thus, in general, the schedulability region of the HRDF scheduler can be significantly smaller than that of EDF depending on the specific parameters. However both have the similar behaviour for the case of simulataneous greedy arrivals.

Fairness

Fairness defined for the rate provisioning schedulers refers to distributing spare bandwidth to the backlogged queues in the ratio of their weights (GPS). However in the context of delay provisioning schedulers, this concept needs to be re-examined. For a connection requiring a delay bound, the average rate is not a significant metric. Fairness in this case needs to take in account the cell-by-cell QoS provisoining nature of this scheduler.

Consider a set of queues backlogged at a particular time. Each head-of-the-line cell has a deadline associated with it, and the waiting time of the cell is calculatable. In order to compare across the queues, the figure of relative delay, i.e., the ratio of wainting time to the local delay bound is a very intuitive metric.

The HRDF scheduler schedules the queue with the maximum relative delay, thus it attempts to minimize this metric. In this notion of fairness, the HRDF scheduler is more fair than the EDF scheduler which will choose the queue with the minimum deadline. However the schedulability region of the EDF scheduler is larger and thus, looking from the viewpoint of having no QoS violations, the EDF scheduler is better as it can accomodate more connections.

4.7 Summary

This chapter proposed a hierarchical scheduling discipline in order to satisfy diverse QoS requirements. The abstract notion of urgency is also introduced. The definition of urgency is different for different QoS requirements. The HRDF scheduler was then introduced and this was shown to serve as a delay or a jitter bounding scheduler. A fluid model analysis of the scheduler is made and the schedulability region under flows constrained by the dual leaky bucket model (used by the ATM UPC functions) is calculated. This region is shown to be strictly less than the region of GPS. However, it is the same as the usable region (i.e., the region in which the weights for different connections are easily calculatable from their QoS requirements) of GPS.

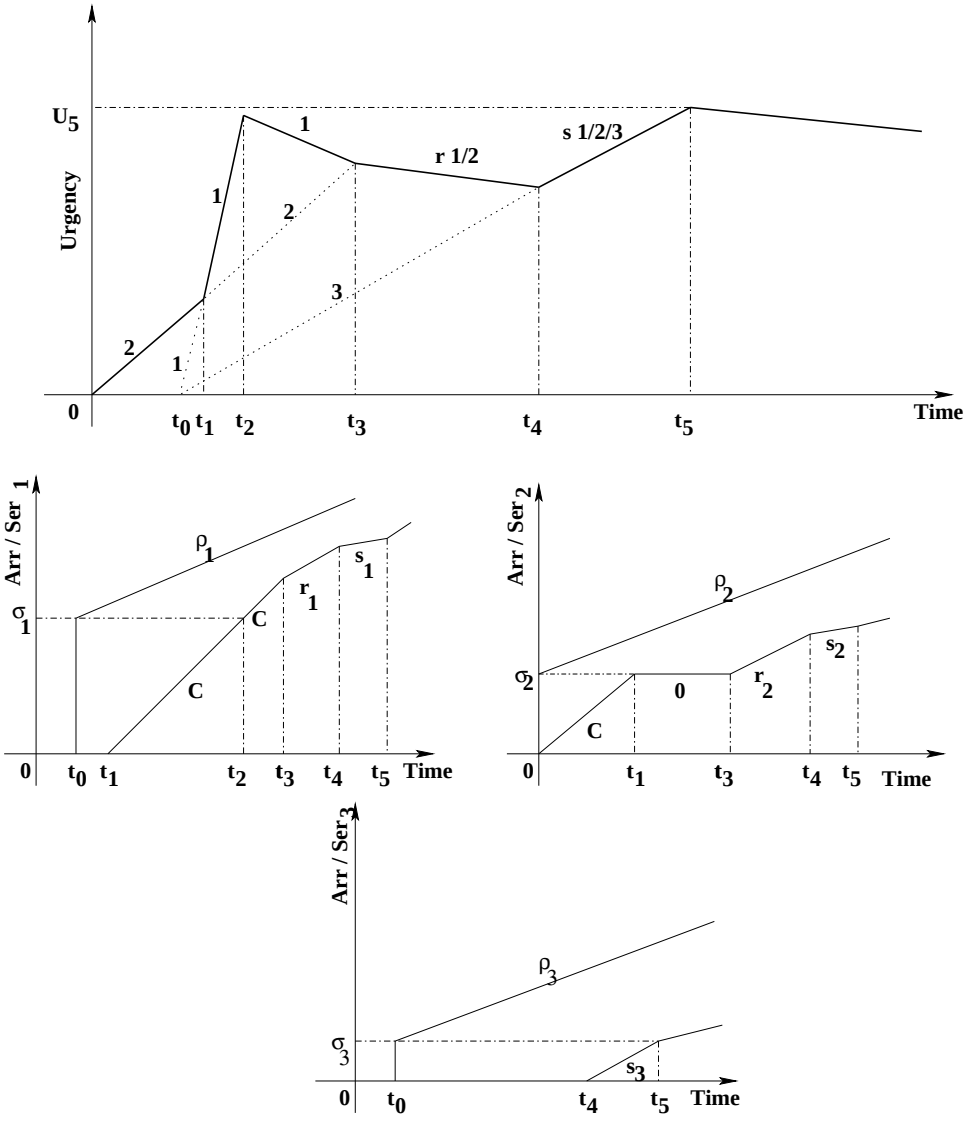

Figure 4.5: Urgency and Arrival / Service curves for the case of connection 2 arriving at time 0 and connections 3 and 1 arriving simultaneously later.

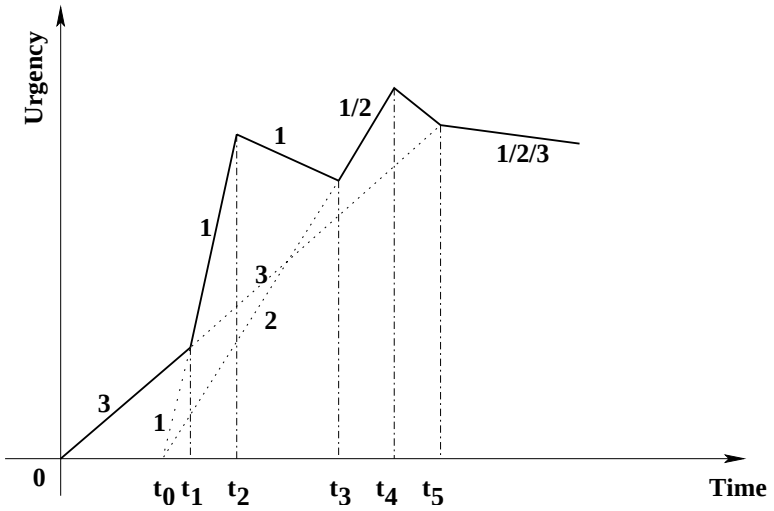

Figure 4.6: Urgency curve for the case of connection 3 arriving at time 0 and connections 3 and 1 arriving simultaneously later (case 2).

Chapter 5

Implementation of the HRDF Scheduler

5.1 Platform

The HRDF scheduling algorithm proposed in this thesis is implemented in a prototype ATM switch. The aim is to conduct experiments on a real switch and correlate the results with simulation of similar scenarios, and also with the results of the analysis performed in the last chapter.

The platform for this implementation is the PSAX-1250 ATM Access Concentrator manufactured by Lucent Technologies. This was a joint collaborative effort between the University of Maryland and Lucent Technologies. The PSAX line of products from Lucent is one of the leaders in the Access Concentrator market. The PSAX-1250 hosts a 1.25 Gbps backplane and 12 Input/Output (I/O) card slots. The 'brain' of the switch is the Main CPU card which implements all the signaling protocol stacks, the user interface, and has the ability to configure the I/O cards as well as administer the connections on the card. This switch supports a wide variety of I/O cards, including many serial cards, DS3, E3, DS1, E1, OC3, and OC12. The channelized DS3, OC3, or the OC12 are usually used as the uplink to the WAN network.

The implementation of the proposed scheduling algorithm is performed on the

OC3-APS I/O card. This is a 155.5 Mbps line rate module that carries 353207.55 ATM cells per second. This card has an on-board CPU and a programmable FPGA. The card performs the SONET link layer functions, VP/VC lookup of cells, and UPC policing functions in the hardware. The queueing and de-queueing of cells is one of the tasks performed by the code for the on-board CPU (firmware). The cells are then forwarded to the backplane, to be delivered to the destination slot. The queueing discipline and the scheduling algorithm used in the card is a Lucent proprietary algorithm called AQueMan (Adaptive Queue Management). There are a fixed number of queues (10), and the CBR queues have the highest priority. The priority of the VBR queues is adaptive, based on the buffer depth of the queues. The UBR connections have the lowest priority. There is no implementation of the ABR service on this switch. As is observable, this queueing discipline does not take into account the delay or loss statistics desired by a connection. Moreover the desired bandwidth of the connection is also only for CAC and UPC functions. The scheduler has no knowledge of the bandwidth requirement of a connection. This algorithm provides a per-class QoS differentiation, but it cannot guarantee per-connection QoS.

The queueing discipline in the card is converted to the one proposed in the thesis. There were a number of modifications required in the firmware, as detailed in the next section. Experiments on the modified I/O card were performed using an Adtech-4000 Broadband Tester equipped with OC3 Line-Interface and Generator-Analyzer modules. The Adtech is capable of producing many traffic patterns, and in addition, a UPC can also be applied to the traffic stream before the traffic leaves the system. The Analyzer module is capable of displaying various statistics of the received stream. Real time graphs of many parameters are also available from the module.

5.2 Firmware Modifications

A number of firmware modifications were implemented for the purpose. The main issues and the associated changes are elaborated upon.

5.2.1 Computing power

The architecture of the I/O modules for this switch has an emphasis on flexibility and programmability. Thus it is not a ASIC based design, and the hardware and firmware share many functions of the card. The firmware controls the queueing and dequeueing of cells in both directions, i.e., to and from the backplane. Limitations on processor speed and memory bandwidth restrict the number of CPU cycles available for computation per cell time. This prohibits the use of a very complicated scheduler on a large number of queues.

Thus to implement the proposed scheduler, the output line rate had to be reduced to a quarter of OC3, giving nearly four times as much time for processing a cell. This was performed by running the dequeueing algorithm every forth slot and inserting null cells in the remaining slots. The card was still operating at 38.8 Mbps = 88301 ATM cells/s. The input line rate was not changed.

5.2.2 Line-rate clock and timestamping cells

The main loop of the card runs faster than the OC3 line rate as the bandwidth coming in from the backplane is much higher. Thus the loop clock of the firmware is not synchronized to the line clock. The line clock timer was synchronized to the number of cells being sent to the hardware rather than the main loop timer. For the implementation of the scheduler proposed, it is necessary to timestamp the cells with the time they come into the queueing engine. An unused 32 bit field was used

for this purpose. A 32 bit counter has a loopover time of about 13 hours, which is many orders of magnitude larger than the cell waiting times. This timestamping is performed only if a connection belongs to the delay or jitter schedulers, as this operation involves a memory operation on the cell memory which is relatively very slow.

5.2.3 Connection parameters management

The console for the connection addition in the main CPU card does not take the desired QoS parameters (CTD, CDV) as inputs. However there are ten different queueing types available. These were mapped to connections that need a rate guarantee, a strict delay guarantee, a jitter guarantee, and those which did not need any kind of guarantees (UBR). The parameters required by these connections are then indirectly passed through the VCI field of the connection. During the experiments, the connections are setup with the VCI mapped to the desired QoS parameter.

5.2.4 Queueing of cells from the fiber

In an output-buffered switch with backplane speed as high as the total input from all the cards, there should ideally be no need for the input queueing mechanisms. However, although the backplane can handle very high rates of data transfers, the I/O cards may not be capable of sinking in cells at that rate. Every I/O card has its own capacity to sink cells from the backplane. This sometimes necessitates queueing on the ingress side (coming from the fiber and going to the backplane) also. In the experiments, similar cards will be used and the backplane would not be carrying any other traffic. There would be hence no ingress queueing, and the

queueing and dequeueing mechanisms on the ingress side are not modified.

5.2.5 Queueing of cells from the backplane

The most significant change in the firmware is the queueing and scheduling of cells in the egress direction. There is support for 32*1024 connections per I/O card in the system. A per-VC queueing mechanism would be very demanding on memory and a scheduler for 32k queues would be very demanding on CPU cycles. It is also observable that if two connections need the same delay, then the order of departure of cells would be the same if the connections were queued in different queues or if they shared the same queue. This is so because the priority curves of cells of these connections would be parallel and will never be able to intersect.

The granularity of QoS provisioning can not also be made very small. As an example for a constraint, the delay guarantees have to be multiples of the cell time (2.83 μs). The maximum delay guarantees expected by connections are of the order of the inter-cell times of the lowest data rate connection supported. Usually the lowest data rates supported are of nearly DS0 bandwidth which is about 170 cells/s. At the OC3 line rate, this is about 2065 slots per cell (inter-cell time). Thus taking the above considerations in regard, it was chosen to granulate the maximum delay supported into pre-defined QoS classes. The supremum of the maximum delay supported is chosen to be 1100 slots per cell. The reason is that if the DS0 connection desired a delay bounded QoS, the delay bound has to be less than the inverse of the cell rate (otherwise the connection can be scheduled as a rate connection). Assuming that the delay bound needs to be tighter than half of the inverse, bounds upto 1032 slots/cell are required. The granularity of the QoS classes is chosen as follows:

1. minimum is 4 slots/cell, next level is 10 slots/cell (1 queue)

2. 10 slots/cell to 200 slots/cell in intervals of 10 slots/cell (20 queues)

3. 200 slots/cell to 600 slots/cell in intervals of 20 slots/cell (20 queues)

4. 600 slots/cell to 1100 slots/cell in intervals of 50 slots/cell (10 queues)

Thus there are a total of 51 delay bounded QoS classes and correspondingly 51 queues for the delay scheduler. The inverse of the delay bound and the urgency of the head of the line cell is stored for each queue. There is also a single bit status (that can be accessed in blocks of 32 bits) for every queue to indicate if it is empty or not. This speeds up the processing significantly.

The rate connections cannot be coupled with each other and have to be queued in a per-VC manner. In this study, our main aim is to understand the behavior of the QoS sensitive connections and so a very sophisticated rate scheduler is not implemented. In general, any rate scheduler that guarantees average bandwidth can be used, with the notion of the urgency defined. The jitter sensitive connections are treated the same as delay sensitive, but with a very short delay bound.

Since the jitter sensitive connections are the most QoS sensitive of all classes, the higher layer scheduler gives the highest priority to them. The rate connections are at the lowest priority, and the delay connections are in the middle. If there are no cells of any of these classes, the UBR (no guarantee) cells are dequeued.

5.2.6 Synchronization of start of bursts

As observed in the analysis section, the scheduler is sensitive to the times at which the connections become greedy. It is possible to synchronize the start of bursts of more than one connections using the Adtech but it is not possible to delay the

burst of one of them by a certain number of slots. However the queueing engine in the card can delay the burst by dropping the first k cells. The number of cells to be dropped (k) is assumed to be the VPI of the connection. In the generator module, k is also added to the desired σ of the connection to keep the parameters consistent.

5.3 Simulation Program

To correlate with results of the experiments on the switch, a small simulation program was also used besides the theoretical analysis. The program exactly emulates the scheduling architecture and algorithms given in the previous chapter. The connections can be specified by the vector $(\sigma, \rho, C, \delta, \tau)$ where τ is the time when the connection becomes greedy. The step size of the computation is controllable, i.e., the granularity of service provisioning can be programmed. Using a step size of $\frac{1}{1000}$, results are found to be close to the theoretical fluid analysis. If a step size of 1 is used, the results are closer to the results from the experiments on the switch. The outputs of the program are various delay and urgency statistics. Data on arrival profile, service profile for delay and GPS schedulers, and urgency profile is generated as a function of time for all the connections, and can be used for plotting.

5.4 Experiments on the Switch

Experiments were conducted with different QoS requirements and varied policing parameters. The maximum delay in the queue was monitored, using peak-to-peak 2-point CDV reported by the Adtech. This is the difference between the

maximum and the minimum CTD for the connection. Since there was no other traffic on the switch, the minimum CTD is the sum of the propagation time and the processing time. Hence, this difference should accurately reflect the maximum delay experienced by the cells of that connection. The CTD of a cell is measurable using a special cell (called Test-32 cell) provided by the Adtech Generator-Analyzer module. This cell is timestamped by the Generator at the point of egress to the fiber and also by the Analyzer at the time of ingress from the fiber.

From the analysis in the previous chapter, the expected maximum cell delay for only one connection can be derived (keeping C_i as a parameter). The delay can be expressed as $\frac{\sigma_i}{C_l} - \frac{\sigma_i}{C_i}$, where C_l refers to the line rate of the card which is a quarter of C, the OC3 line rate. Following are the details of the experiments and the results of 10 instances of the experiment:

1. $\sigma_1 = 10, \;\; \rho_1 = 0.20 * C, \;\; C_1 = C$

   Theoretical maximum delay: $84.94\mu s$

   From simulations: $84.93\mu s$

   From experiments: figure 5.1.

2. $\sigma_1 = 30, \;\; \rho_1 = 0.20 * C, \;\; C_1 = C$

   Theoretical maximum delay: $254.8\mu s$

   From simulations: $254.81\mu s$

   From experiments: figure 5.2

3. $\sigma_1 = 100, \;\; \rho_1 = 0.20 * C, \;\; C_1 = C$

   Theoretical maximum delay: $849.36\mu s$

   From simulations: $849.36\mu s$

   From experiments: figure 5.3

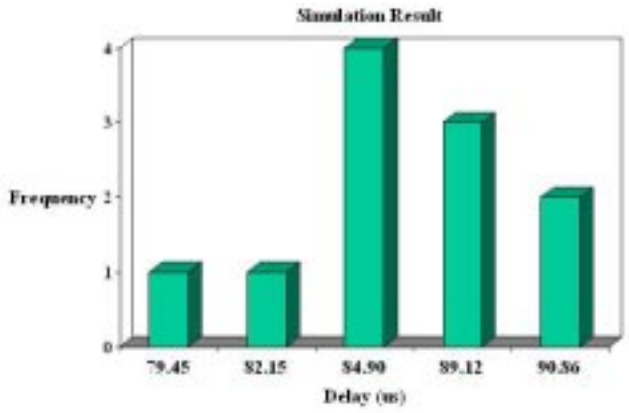

Figure 5.1: Delay histogram for one connection with burst size 10.

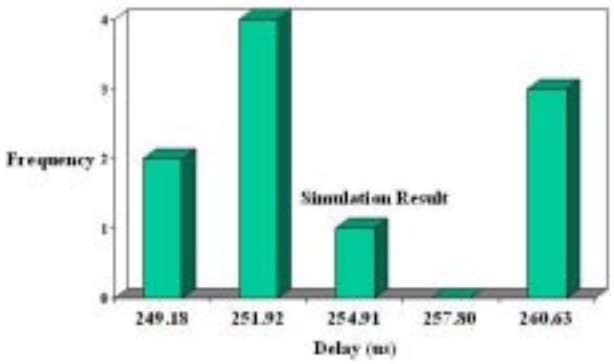

Figure 5.2: Delay histogram for one connection with burst size 30.

For experiments with larger number of connections, the Adtech had to be calibrated to provide the exact cell departure pattern required. This was done by observing the capture in the Diagnostic Loopback mode. Consider the following

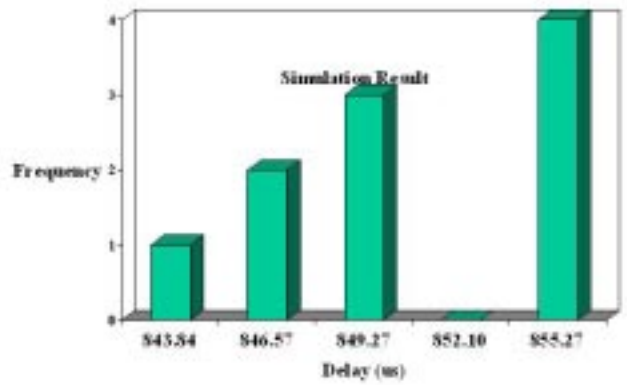

Figure 5.3: Delay histogram for one connection with burst size 100.

scenario: two connections with $\sigma_1 = 20$, $\rho_1 = 0.24 * C = 0.96 * C_l$, $C_1 = 2 * C_l$ $\delta_1 = 30$ and $\sigma_2 = 50$, $\rho_2 = 0.01 * C = 0.04 * C_l$, $C_2 = 2 * C_l$ $\delta_2 = 600$. The maximum delay numbers obtained from the simulation are $D_1 = 393us$ and $D_2 = 7655us$. A histogram for the delay values obtained in 25 instances of the experiment is shown in figure 5.4. It is observable that the delay remains within $\pm 1\%$ of the prediction in the case of one connection. With two connections, the error is slightly higher, within $\pm 1.5\%$ of the simulation. Even in experiments with 20 connections, the error was bounded by $\pm 2\%$ of the simulation results. This is very accurate because the number of FIFOs in the path of the data is nearly 15, and the scheduler only controls one of them. The rest function in an asynchronous manner, thereby introducing many uncontrollable delays.

The performance of this implementation of the HRDF scheduler is restricted by many factors, mainly, the serial implementation of the core in software and the need write and read a timestamp for each cell in the memory. Therefore, the

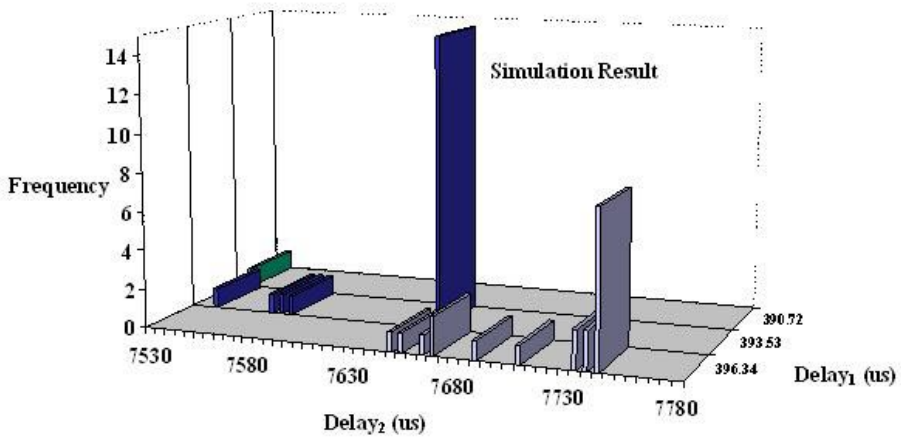

Figure 5.4: Delay histogram for two connections with burst sizes 20 and 50.

maximum number of connections supported is 20 and for more connections, there are cell drops due to the backplane FIFO filling up. A parallel implementation of the algorithm in hardware would be much more efficient and would be able to support orders of magnitude more connections. In the next chapter, a review of the state of the art in scheduling technology is made, in order to understand the capabilities of the current hardware.

5.5 Complexity

The implementation of the HRDF scheduler for N queues involves N additions and a maximization in one cell time. In a hardware implementation, additions can

be executed in parallel in one cycle using N adders. To find the maximum of the results of these additions, N comparisons are made in the next cycle, $\frac{N}{2}$ in the next and so on. This process takes $log(N)$ cycles and uses nearly $2N$ comparators. Since $O(N)$ adders and comparators are not a limiting resource in hardware, the main limitation is that of using $(1 + log(N))$ cycles $(O(log(N)))$.

In an OC48 link with discrete delay bounds (like the one described in our implementation), similar guarantees can be achieved with about $N = 1024$ queues. This means the scheduler would need 11 cycles. The cell time at OC48 rate is about $177ns$. Nearly half the time ($80ns$) is available for dequeueing, and the rest half is for queueing process. At typical hardware speed of 200 MHz ($5ns/cycle$) there are $16cycles$ available for dequeueing, which would suffice for implementing a HRDF scheduler.

As a comparison, take an EDF scheduler. At every cell time, again a maximization of N numbers needs to be done. Since the numbers do not change every cycle, an ordered list can be maintained. Thus an insertion into the ordered list is required, which will take $log(N)$ cycles using a binary search. This is very similar to the number of cycles used by the HRDF scheduler. Consider also a GPS scheduler implementation [40]. Besides a maximization step that involves $O(log(N))$ cycles as in the EDF case, it requires the computation of the system-potential function. Although this computation is very complex for the ideal case, there are several approximations proposed to do it in $O(1)$ complexity. Thus the hardware implementation complexity of all the three schedulers is $O(log(N))$.

Chapter 6

State-of-the-Art in Scheduling

This chapter presents a review of the state-of-the-art of the scheduling technology. There are two categories of companies which make products related to ATM and use scheduling algorithms. The first category is of the vendors who produce switches, either core or access. The implementation of the scheduling algorithms is very diverse amongst different products. It can be done in different architectures (Input Queued, Output Queued, or mixed Input-Output Queued) with the use of different technologies (in hardware or software). The second category comprises of vendors who make Network Processors. These are highly specialized processors which handle communication protocol stacks. Recently, some companies have introduced products which handle both the physical layer protocol and the ATM layer functions on single or multiple chips.

The most important consideration in the industrial environment is the cost of development and production of a product. The simplicity of implementation and the scalability of the algorithm are the prime factors that are to be considered. Design of algorithms which are simple, scalable and yet powerful enough to satisfy the user requirements is invariably the challenge. In this review, the focus is to find the level of complexity in the state-of-the-art algorithms.

6.1 Core and Edge Switches

6.1.1 Marconi Corp.

One of the largest core switch manufacturer, Fore Systems (currently a division of Marconi Corp.), has a variety of network modules for their ASX line of ATM switches. The interface cards for these switches, called Network Modules, come in various revisions. The most advanced "series E modules" offer per-VC queueing and shaping [41]. These modules also offer a second tier of per-VP shaping. Fore offers these capabilities for upto 32k connections, i.e., 32k connections can be individually shaped. Fore uses ASICS in their network modules to implement these algorithms. Older network modules from Marconi have a per-Class WRR algorithm or priority scheduling. The figure 6.1 shows the data path in case of a series E network module with traffic shaping.

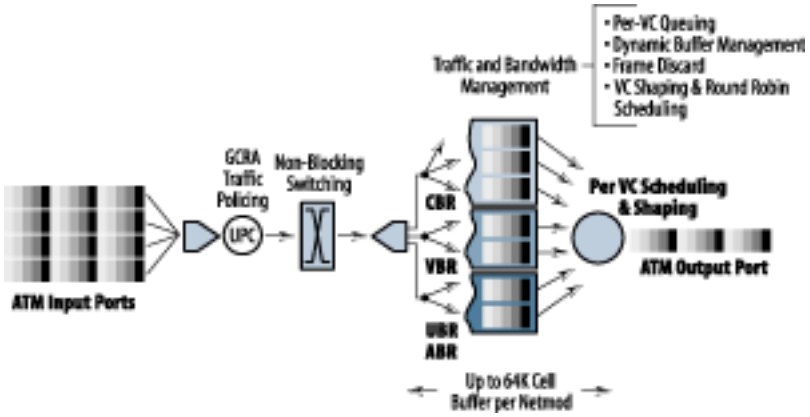

Figure 6.1: Scheduler in series E network modules from Marconi.

6.1.2 Cisco Systems

Cisco systems produces edge ATM switches in its MGX and Catalyst line of products. It also produces small core switches in the LightStream family and ATM in-

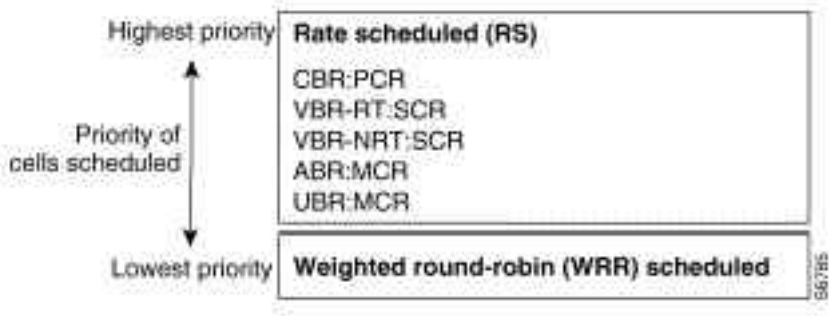

Figure 6.2: Priority order of queues in Cisco products.

terfaces for their IP routers. The MGX edge concentrators support sixteen static priority classes and per-VC traffic shaping for upto 4k VCs in one module [42]. The Catalyst switches support WRR on a small number of queues [43]. The LightStream core switch also has static priority queues, and also supports traffic shaping [44] for frame based protocol to ATM interworking. This is required because the segmentation of a frame usually produces more than one ATM cell and the transmission intervals of these cells needs to be regulated by the contracted ATM PCR. The ATM interfaces for the high end IP routers provide for class based WFQ and not per-VC WFQ [45]. From [46] it can be inferred that all the Cisco products provide per-Class WFQ or WRR and only per-VC traffic shaping in rare cases. The priority order of various queues in Cisco products is shown in figure 6.2. This is an example of a simple hierarchical scheduler.

6.1.3 Alcatel

The OMNI family of aggregation switches from Alcatel [47], support Programmable Bandwidth Queueing (PBQ) which is very similar to the WFQ algorithm. It uses a WRR on a fixed number of queues (8) with a programmable 'burst enable'

mode. Disabling the burst mode imitates the traffic shaping capability. The Alcatel 7420 Multi-service Edge Services Router [48] offers per-Class WFQ and priority queueing. This switch also has eight outbound queues only. Alcatel apparently does not offer any per-VC traffic management function on the output side.

6.1.4 Lucent Technologies

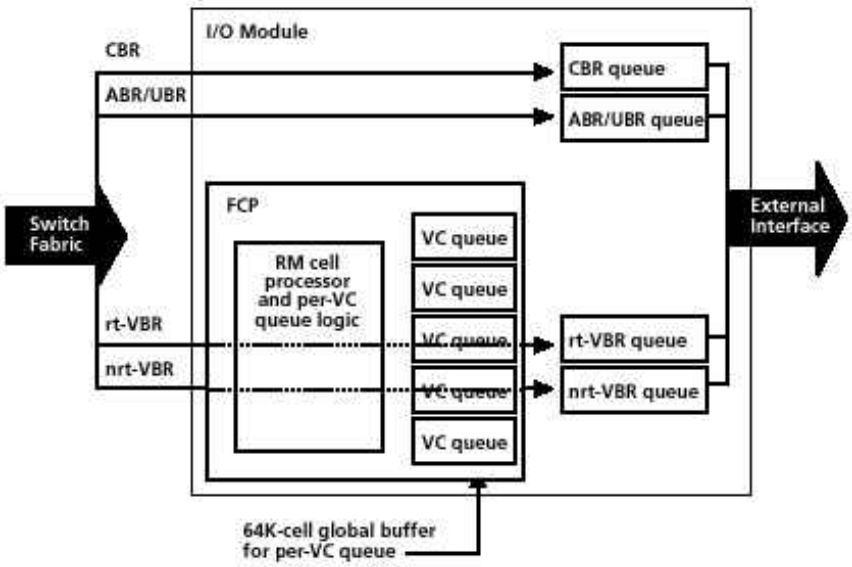

Figure 6.3: Queueing structure in a Lucent CBX switch.

The CBX family of Multi-Service WAN switches from Lucent Technologies offers per-VC or per-VP traffic shaping capabilities for VBR class of traffic [49]. It provides static priority scheduling on a per-Class basis. The queueing structure in the CBX switch is shown in figure 6.3. The PSAX family of Multi-Service Media Gateways [50] possess the patented AQueMan algorithm that has fixed number of queues (10). It schedules cells on dynamic priority discipline in which the priority of the VBR queues is calculated in real-time based on the depth (number of cells in the queue) and age (time since the last cell was dequeued) of a queue. The newer

I/O modules in the PSAX family offer the traffic shaping capability for upto 32k connections. Some I/O modules also offer Virtual Interfaces which shape a bundle of connections. The GX family of Multi-Service WAN switches from Lucent [51] offer per-VC or per-VP traffic shaping on their ATM I/O modules. They have per-Class strict priority queueing at a higher level.

6.1.5 Nortel Networks

Nortel's product line includes the Passport family of Multi-service platforms [52], [53] equipped with core switches and edge access switches. These switches have per-VC queueing and a per-Class WFQ scheduler. There are eight classes of traffic defined. The switches also support traffic shaping (inverse dual leaky bucket) at UNI interfaces.

6.1.6 General DataComm

The APEX family of switches from General DataComm are positioned in the Multi-service Access Concentration and Enterprise backbone markets [54]. These switches offer traffic management using Multi-tier Shaping algorithm [55]. This algorithm is capable of shaping a VC to a specified bandwidth and then shape a VP also in a hierarchical manner as shown in figure 6.4 . This ensures that the output traffic is in conformance with any contracts for a particular VP and the traffic would pass through a VP policer (which are common on core switches). Such a feature is useful for traffic aggregation, for example, a user can be sold an entire VP and then after exiting from this switch not only the individual connections (VCs) would be in conformance with the contract, the entire pipe (VP) would also be in conformance. The switch uses per-VC queueing to implement this feature.

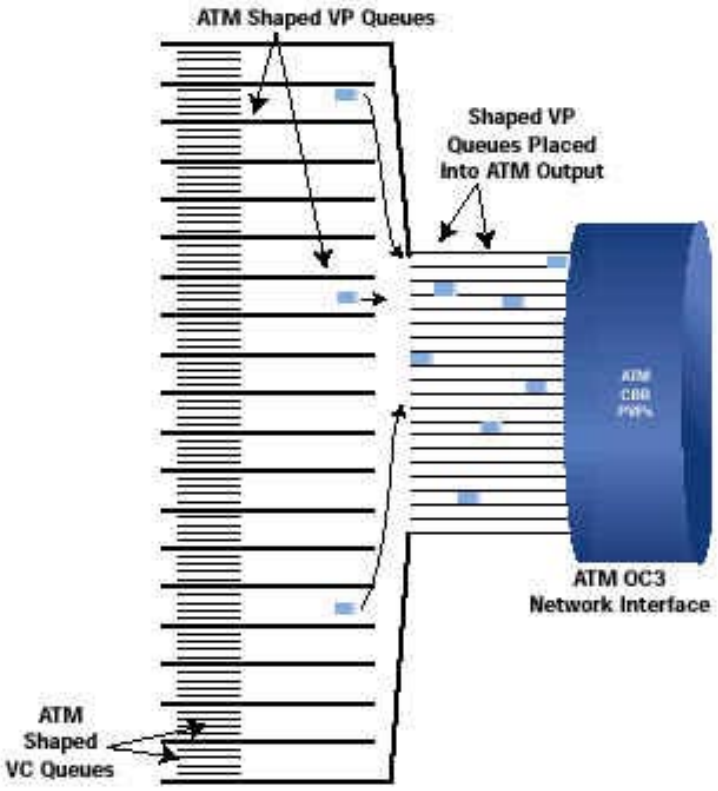

Figure 6.4: Multi-tier shaping in GDC APEX switches.

6.2 Network Processors

Many chip makers have introduced a class of highly specialized processors that implement numerous functions of a protocol stack. For ATM, chips for the physical layer were available for a long time, and recently there are chips in the market that implement the physical layer protocol (like SONET), and handle the ATM cells queueing and dequeueing to the switching fabric. Since this is the latest technology in the fastest ASICs of today, it is expected to be much more powerful than the traffic mangement functionalities reviewed above.

6.2.1 Globespan Inc.

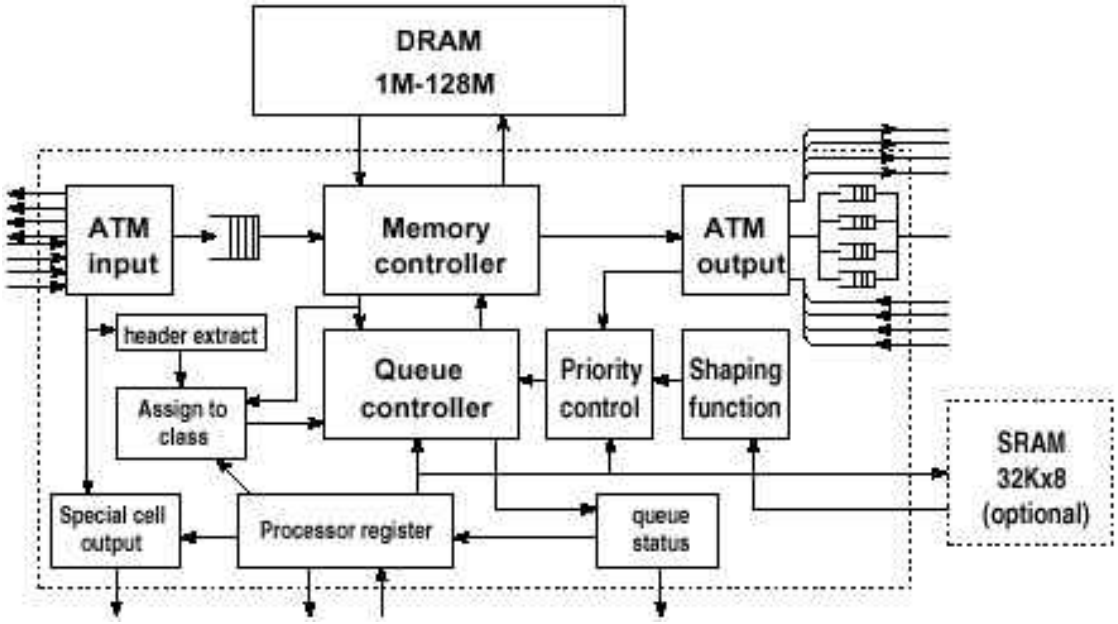

Figure 6.5: Functional block diagram of the SHAP4 processor from Globespan.

The AteCom division from GlobeSpan Inc. [56], offers the ATM SHAP3/4 processors which have 32 queues and an inverse dual leaky bucket traffic shaping per group. It also supports dynamic round robin queue priorities between groups. Inside each group, there are three traffic classes. The medium priority VBR traffic class can be traffic shaped in a hierarchical manner. A block diagram of this chip in figure 6.5 illustrates the typical data path inside a traffic management co-processor chip. It uses external DRAM memory for cell storage and faster external SRAM memory for maintaining various data structures.

6.2.2 Conexant

Conexant [57] offers the MXT 4400 traffic stream processor and the PortMaker processor software. The package offers 32k full duplex connections, with per-VC

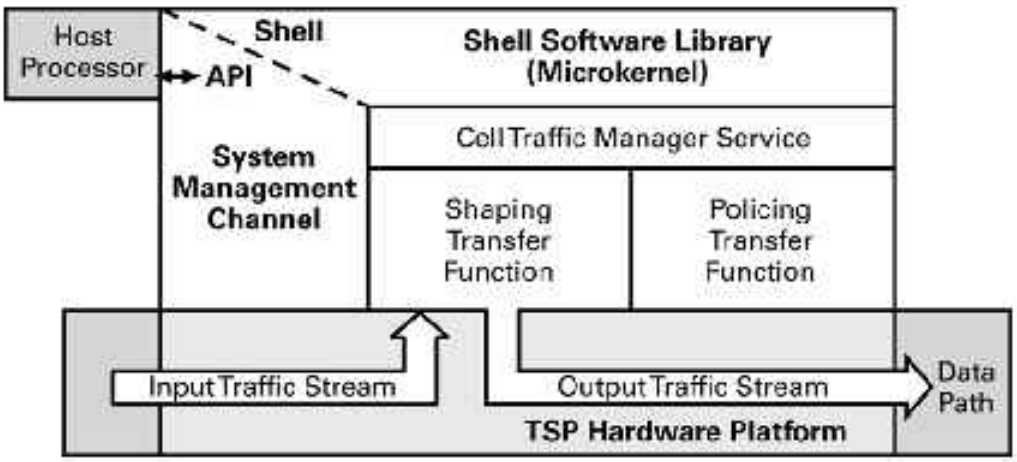

Figure 6.6: Schematic of the MXT 4400 chip from Conexant.

traffic shaping and VP tunnel shaping for upto 128 VP tunnels. This is mainly a AAL5 SAR package and the shaping is in reference to that also. The scheduler in this package is of WRR kind. The MXT 4400 chip is meant to be used with a host processor and is completely programmable. A schematic of this chip is shown in figure 6.6.

6.2.3 Transwitch Corp.

The Cubit [58] and the Aspen [59] products from Transwitch Corp. offer very simple per-Class (4 classes are defined) strict priority scheduling for ATM layer. Besides UPC, Transwitch does not offer any advanced traffic and congestion management mechanisms. Data path in the Cubit product from the CellBus (backplane) to the optical line (UTOPIA interface) is shown in figure 6.7.

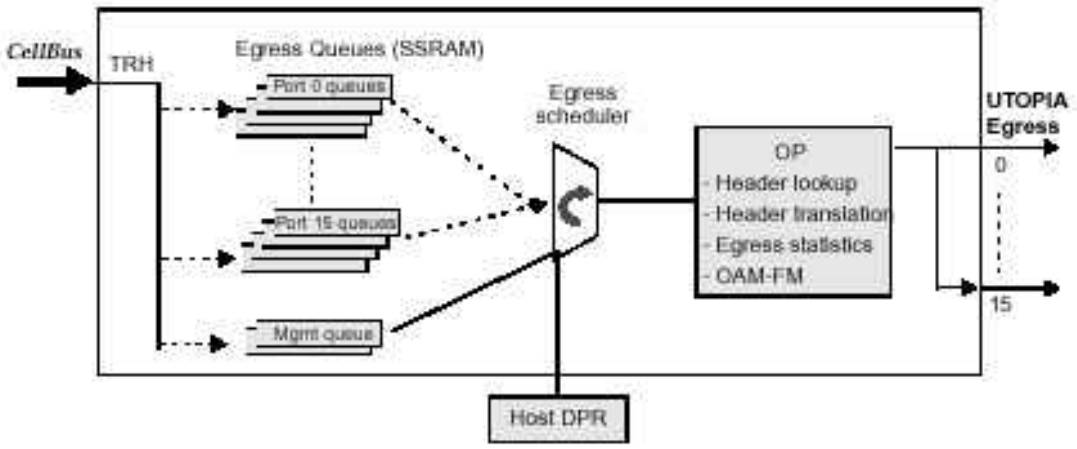

Figure 6.7: The data path in the Cubit chip from Transwitch.

6.2.4 PMC-Sierra

PMC-Sierra, a reputed embedded products manufacturer, has an array of products for ATM physical layer, ATM switching layer and AAL layers. The Apex chip [60] has ATM traffic management capabilities. It supports 64k per-VC queues, and 4 CoS queues. The per-Connection scheduler that feeds into per-Class queues is WFQ and the per-Class scheduler is strict priority with minimum bandwidth reservation. On some special ports, per-VC rate shaping is also offered.

6.2.5 LSI Logic Inc.

The line of ATM products from LSI Logic Inc., includes the ATMizer II [61] which implements a sophisticated scheduler in its ATM Processing Unit. It supports 64k VCs and 6 priority classes. The scheduler supports 4 calender queues which themselves can be operated in a flat mode (all cells are treated same) or priority mode (cells differentiated based on the priority classes). The calender queue implementation can be used for traffic shaping or a deadline based scheduler. The calender queue is a well known data structure used for shaping and scheduling. For every

cell slot in future, a list of the cells that need to be dequeued in the time slot is maintained. Calculation of the time to dequeue a cell gives various flavors to the algorithm. Since only a single cell can be dequeued in a certain time slot, the way a scheduler handles multiple cells waiting to be dequeued at the same time also accounts for many variations in the algorithm. This particular implementation transmits the first cell in the list of a given time slot and then moves the list to the next time slot. The list is maintained according to the priority class of the cell.

6.2.6 Acorn Networks

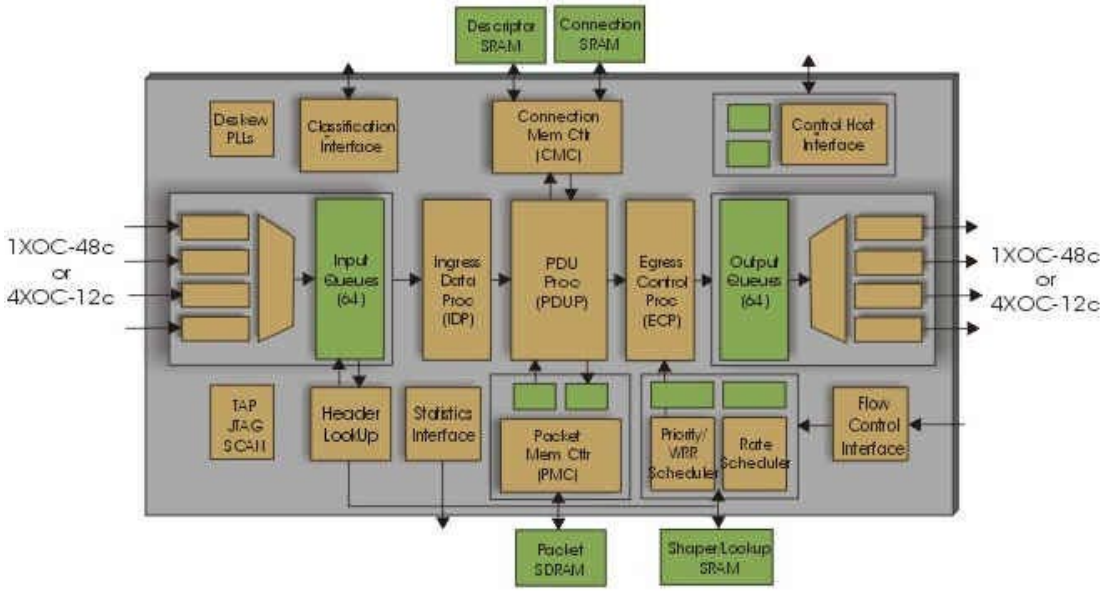

Figure 6.8: Functional block diagram of the genFlow chip from Acorn.

The genFlow product from Acorn Networks [62] is OC-48 Multiprotocol Traffic Management Coprocessor. It provides per-Connection queueing for 64k connections using a sophisticated scheduler that organizes traffic flows into 1k rate-based scheduled queues, 512 priority-based queues, and 512 weighted round robin queues.

The main strength of this processor is in its ability to handle multiple protocols like IP, MPLS, ATM, PPP, Frame Relay and User defined proprietary formats. A block diagram illustrating the interaction of main functional blocks in this chip is shown in figure 6.8.

6.2.7 ZettaCom Inc.

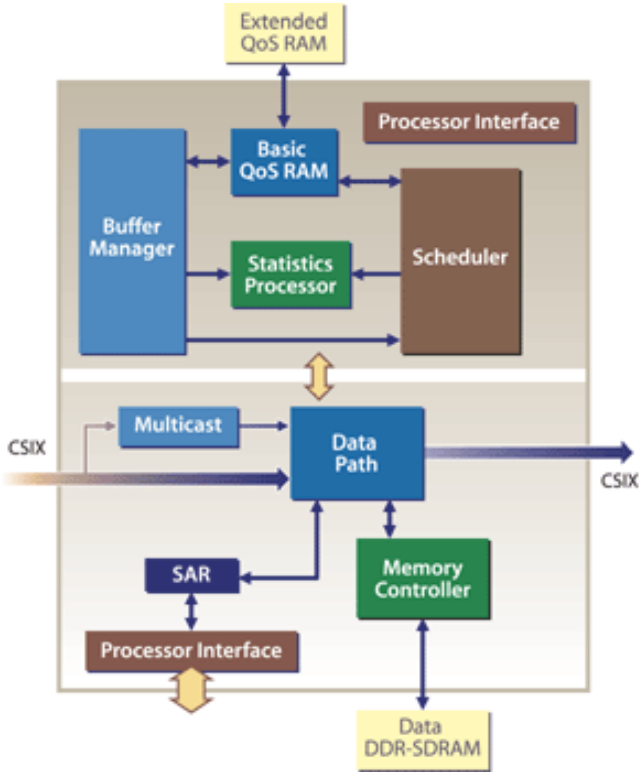

Figure 6.9: Functional block diagram the ZEN-QM queue manager chip from Zettacom.

The ZEN Multi-Protocol Processing Family from ZettaCom Inc. [63] includes the ZEN-QM Traffic Management that support quality of service differentiation for any data protocol (IP, MPLS, ATM etc.) at 10 Gbps speeds (OC-48). This chip offers per-flow, per-class, and per-logical port rate shaping capability. Per-class

priorities are builtin and bandwidth distribution can be dynamically weighted.

6.2.8 AMCC Inc.

The product line of MMC Networks division of AMCC Inc. includes the nPX5700 10 Gbps Traffic Manager Coprocessor [64] that has per-flow queueing mechanism with support for strict priority, weighted fair queueing, weighted round robin and minimum / maximum bandwidth control. This chip supports hundreds of thousands of queues and also implements the admission control policy in hardware.

6.2.9 Vitesse Semiconductor Corp.

The most powerful processor, PaceMaker 2.4, is offered by Orologic subsidiary of Vitesse Semiconductor Corp. The chip has per-VC queueing for 256k connections, a dual leaky bucket shaper per connection, and an Earliest Deadline First scheduler to handle OC-48 ATM traffic [65]. This is the only device that can explicitly support delay and rate guarantees. The number of connections supported is also the largest in this device. PaceMaker offered the first silicon implementation of the EDF algorithm, a quantum leap in QoS provisioning tools. Beta version of this chip came out in June of 2000, and its general availability was announced in May 2001 [66].

6.2.10 Others

Several other startup companies, namely Azanda Network Devices [67], Silicon Access Networks [68], and Bay Microsystems [69] claim to offer advanced traffic management features on silicon although they had not revealed the details of their product lines at the time of this writing.

6.3 Summary

The core and edge switch manufacturers offer per-class Weighted Fair Queueing or Weighted Round Robin scheduling, and advanced traffic shaping capabilities. Traffic shaping is necessary in Edge switches because the policing is usually turned on in core switches. If the traffic is not shaped at the egress of the edge switch, it is vulnerable to policing because of burstiness introduced in the traffic by the edge switch or the equipment before it. This is not a good network design, as the traffic should be policed only once at the ingress of the network (i.e., at the edge switch). The downstream switches should attempt to provide the desired QoS to the connection. However, the internet is a network of many networks owned by different providers, and it is difficult to design optimally it such that the connections are policed only once. Consequently, the traffic shaping function plays an important role in avoiding unwanted traffic drops. Some service providers sell bandwidth to customers in the form of VPs (a bundle of VCs) and then core switches are programmed to perform per-VP traffic policing. In that case, it is necessary to perform per-VP (or any other defined bundle of connections) shaping at the edge of the network. Some of the switches offer per-VC or per-VP shaping but a few vendors also offer the multi-tier shaping capability. In addition to traffic shaping at the edge, QoS provisioning is required in the core also. Per-class WFQ/WRR are insufficient mechanisms to provide distinct QoS guarantees to every connection. However, this is the best that the switch manufacturers are offering at present.

The vendors marketing Network Processors or Traffic Management Coprocessors offer more capabilities in their chips. Per-VC and / or per-VP inverse dual leaky bucket rate shaping is offered in many products. Most products, however, are still based on a per-class scheduler which is either WFQ or WRR, and the number

of queues (classes) is usually a small number. There is only one product in the market that implements the EDF (Earliest Deadline First) scheduler in silicon. This chip also offers per-VC dual leaky bucket rate shaping. The processing capabilities of this chip are humongous and it supports the largest number of connections among the products reviewed. Another important step in Network Processor capabilities is the CAC algorithm implemented in the chip in at least one product. This feature offers even more delegation of functions to the lower level processing units in a switch, freeing more resources from the main CPU unit of the system for higher layer functions like signaling and routing. Since the computational complexity of HRDF in a parallel hardware implementation is approximately equal to that of EDF, HRDF is implementable in todays network processors.

Another significant leap in the Network Processor design is the ability of at least two of the reviewed products to handle multiple protocols like IP, ATM, MPLS, FR or other proprietary protocols on a single programmable chip. This illustrates the point that the concepts in traffic management can be made generic to many protocols and the applicability of these concepts is wider than just the ATM protocol.

6.4 A Digression on QoS

Consider an ATM network of a service provider. Typically there would be sources at the customer premises (like DSL modems, cable modems, POTS modems, T1 lines) feeding to edge switches of the service provider. The provider then aggregates the data and feeds it to the core network using an optical link (typically). The core is a mesh network where high bandwidth fibers connect high speed switches. The traffic would either eventually come out on another edge switch demultiplexing

back to customer premises or it would be fed to peer network of a different service provider.

The ATM network defines a traffic contract between a user and the network. According to the contract, the user's traffic should be bound by the traffic parameters that the network may or may not police using the UPC function. The network in turn agrees to provide the user with the contracted QoS defined by cell latency, jitter and errors. Note that these parameters are independent of the traffic parameters. At the egress of the network, the traffic usually does not retain its original burstiness profile and may need to be smoothened in case the next service provider intends to police the traffic according to the traffic contract (which is perfectly valid). Thus at the egress of the network, the "shaping" functionality is required to make the traffic in conformance with the traffic contract and to remove the burstiness introduced in the traffic profile due to the network elements. Since the user is expected to conform to the contract in the first place and many applications cannot be made aware of this requirement, it is also appropriate to introduce the shaping function near the customer premises, before the policing at the network edge takes place. Thus, if the network is designed in an ideal manner, the UPC and the shaping functionalities in the switches are required at the edge or access part of the network of a service provider, and not in the core switches.

The UPC functionality is almost ubiquitous amongst the various products in the industry. Since many products or product families are targeted at edge and core markets, the feature lists of edge and access products look very similar. A noteworthy point in the above review of the products in industry is that a majority of products support some or the other form of traffic shaping (inverse single or dual leaky bucket). As noted, the need for shaping appears to be in a very niche market only and still the products supporting this are numerous.

In todays internet age, the last mile bandwidth has gone up using DSL and cable modems. The "mp3" culture has introduced the trend of large non-real time downloads. Service providers need controls in their network in order to control bandwidth usage by individual users. Shaping their traffic is the most obvious method of achieving this (UPC can also achieve this, however there is no provision in the standards to define a policer that has a burst tolerance on the PCR bucket also). In fact multi-tier shaping, i.e., shaping each connection to its contract and then shaping the complete pipe of the user (that can include many connections) to the contracted rate of the user is a very attractive feature for use in aggregation. This feature is also offered by many vendors. Since many product families are targeted at edge and access markets (specially network processors), shaping is a very attractive feature to offer. It goes without saying that per connection shaping would require some flavor of per-VC queueing.

Another interesting observation in todays internet is that the policing of traffic is done at various points in the network, even in the domain of the same service provider. For example, it is typical that the core switch would also police the traffic, and thus in most cases requiring the edge switch to shape traffic. One of the reasons for this is the diversity in the equipment used in the internet. Although this is not a good network design, it is prevalent and increases the need for shaping in the network.

The QoS defined in ATM is the set of parameters that the network should be able to provide to the user in a predictable manner. The definition of QoS is very subjective in general and depends highly on the application. Voice applications for example are usually CBR in nature and have stringent requirements on delay and jitter. Most switches usually pass the CBR traffic in the highest priority queue, without using any form of per-connection queueing. It is assumed that the UPC

and the CAC should guarantee the bandwidth to the CBR traffic, as there is no over-subscription involved. In case of VBR traffic, which is usually bursty, there is oversubscription of bandwidth by CAC, allowing for short term congestion in the switch. It is in this case that sophisticated scheduling and per-connection queueing is required to guarantee differentiated QoS to every connection. However in most of the equipment today there is hardly any mention for this provisioning in an explicit per connection manner. There is per-VC queueing, but the emphasis is on shaping. The final scheduling is mostly done per-class, and the number of classes defined is typically small. This seems to be so because the QoS aware applications, like real time streaming video, are not possible today mainly due to the limitations in the last mile bandwidth. As cable modems and DSL brought a quantum leap from the telephone modem, another such quantum leap is bound to happen in near future and it would change the order of magnitude of the bandwidth available to a user. Bandwidth and QoS hungry applications would become more popular then. That is anticipated to be the time when the need for per-VC QoS differentiation would really be realized. At the same time, the need for performance management in the network would also become apparent.

In the products reviewed in this chapter, there was no equipment manufacturer that offered this capability of per-VC QoS provisioning. However, the PaceMaker chip from Vitesse seems to be the most advanced in all regards. It can offer the best QoS provisioning by implementing the most optimal delay bound provisioning algorithm, and also offers per-VC inverse dual leaky bucket rate shaping. This product is however very recent, and definitely paves the path to the future of traffic management. Hopefully in future, the demand for increased bandwidth and stringent QoS would lead to the advent of more products with capabilities of this kind.

Chapter 7

Monitoring of QoS Parameters

The objective of monitoring is to ascertain whether the QoS guarantees negotiated at connection setup are being provided. In the OAM protocol special cells injected in a stream to enable accurate estimation/measurement of QoS parameters in a virtual circuit in order to establish conformance with the QoS objectives agreed upon at connection setup. In case a violation or deterioration is detected, the PM device decides to take corrective actions as explained before.

7.1 Delay and Jitter Measurement

A new QoS measurement protocol is proposed in this section. As defined in the standard [2], the parameter CTD refers to the worst case delay value and the network is obliged to provide the average CTD for the lifetime of the connection less than this bound. The performance objective of the parameter CDV is that the difference in the CTD of any two cells should not be larger than the CDV bound. Similarly, the performance objective of the QoS parameter CLR is that the average cell loss over the lifetime of a connection should not be more than the CLR parameter in the contract.

Based on these performance objectives for delay and loss parameters, a new protocol based on pattern cells is proposed to verify the conformance with the

delay bounds. This new protocol has the flexibility to trade bandwidth overhead with the precision of measurement of delay parameters. For this reason the idea of adaptive sampling [70], [71] is incorporated here. As will be discussed later, the protocol has the advantage of being capable of measuring the CDV, average CTD, cell losses at the minimum bandwidth overhead with the required precision.

7.1.1 Pattern cells

It is proposed to define a new type of performance management forward monitoring and backward reporting cells called "pattern cells". For a particular connection (VCC), time is considered to be slotted according to the the PCR of the circuit. By definition, the source will not inject cells with spacing less than $\tau_p = \frac{1}{PCR}$ sec. Thus it is justified to divide time in slots of τ_p seconds starting from the first cell transmission time, as this is the maximum slot size which will contain at the most a single cell. A cell belongs to the slot in which the larger fraction of it was transmitted. Pattern cells reflect the profile of user cell transmissions since the last pattern cell was transmitted, in the form of a bit sequence that furnishes information about the relative time between cell transmissions. Each bit represents one or more time slots, depending upon the OAM bandwidth overhead that can be incurred on the connection.

A pattern cell can be defined to have the OAM type field as "0010" and the function type field "0010" and "0011" for forward monitoring and backward reporting cells respectively. In the 45 octet payload of the forward monitoring cell, it is proposed to use two bytes as sequence number for identification of lost cells, three bytes reserved for future use and the remaining forty bytes representing the pattern. Let the number of time slots represented by a bit be n $(n \, \epsilon \, \mathcal{I}^+)$ and the desired OAM overhead be h $(0 < h < 1)$. Then,

$$n = \left\lceil \frac{PCR}{h * 8 * 40 * SCR} \right\rceil . \tag{7.1}$$

Now let i be the least integer such that

$$\frac{2^i - 1}{i} \geq n. \tag{7.2}$$

Thus, a block of i bits represents $n*i$ time slots and the number of cell transmissions in these slots can be encoded using the i bits. The relative time of transmission of a cell from the time of transmission of the first cell can always be known within $n*i$ PCR time slots. In the standard I.610, the minimum time for sending OAM cells is every 128 cells, thus giving the best case accuracy of determining a cell's relative position to be within 128 SCR time slots on an average. Consider an example: let $h = 0.5\%$, $\frac{PCR}{SCR} = 10$. Thus, $n = 7$ and $i = 6$. So the accuracy of this protocol is to be able to relatively place a cell within 42 PCR time slots equivalent to 4.2 SCR time slots, and the accuracy of OAM protocol (inserting OAM cells after every 128 cells) would be 128 SCR time slots. In this case, the protocol proposed has a bandwidth overhead of 0.5% and standardized OAM protocol has an overhead of 0.77%. For more accuracy, n can be increased to 2. Then $i = 3$ and the relative position can be determined within 6 PCR slots or 0.6 SCR slots. In this case, the overhead is $h = 1.56\%$. The extreme case would that be of $n = 1$ and $i = 1$, where the positioning can be determined accurately, and the overhead is about 3.13%. The price to pay here is in real time encoding of timing information in pattern cells. Also note that the pattern cells indicate the exact number of cells transmitted using the sequence number field. Thus CLR can only be calculated approximately because lost and mis-inserted cells can not be identified using end-to-end monitoring.

Estimation of CDV as difference in the maximum and minimum measurements, or the variance of the transfer delays of cells gives an advantage because of the fact that the measurements are all relative in the protocol. Thus, an absolute time error T_0 in the measurement of CTD of a cell can be assumed because the clocks at the source and destination will never be synchronized. This error would be reflected in the mean of the delays and thus would not appear in the peak to peak CDV, or variance estimates. This implies that the destination can take the time of arrival measurements by its own clock, calculate the time of departure of a cell using the relative position of the cell and the (known) time of departure of the first cell. Cell losses can also be measured by the difference in the number of cells received and the number of cells transmitted (as known from the pattern cell). Cell transfer delays can also be measured within an additive constant. This constant can be determined by sending timestamped cells at periodic intervals.

OAM cells are bandwidth overhead for the data stream. The number of OAM cells transmitted in the lifetime of a connection should therefore be minimum. The idea of adaptive sampling is to analyze the trend in the gathered data and adapt the sampling rate to the minimum rate required for the purpose. In the protocol described above, the overhead being incurred can be reduced by increasing the value of n and thus i (which in turn makes accuracy worse). Thus using this trade off between bandwidth overhead and accuracy, monitoring can be adapted for either better accuracy or less overhead.

7.1.2 Improvement in clock estimation scheme

The clock parameter estimation scheme in [27] can be modified so that each node in the network estimates the parameters of its neighbors' clock [70]. This can be done with better accuracy as it involves only one queueing delay in the path,

and may be measured accurately using a CBR connection during light loading. Once a node knows the drift equations of all neighbor clocks, remote clock offset from source can be known by propagating it hop-by-hop along the connection path at the connection setup time. Thus, the source can know the model for the remote clock at connection setup time and use it to correct timestamp values in OAM cells from the destination. This allows for measurement of one-way delay, although bottle-neck links can still not be known. This scheme, which uses the performance management OAM cell, neither involves a large time delay to achieve accuracy nor new processing capabilities at switches and can be more accurate than Roppel's scheme.

7.2 Identification of Bottlenecks

7.2.1 Loopbacks

One of the simplest schemes that can be used for this purpose is reverse flooding of loopbacks. The OAM performance management cell is looped back by all the switches in the path, giving the source an estimate of the round trip delays to each node in the path (figure 7.1). The difference in these measurements gives the bidirectional delay on a link (including the delay in associated output buffers).

However, this results in very high overhead and measures a lot of unnecessary parameters and still suffers from the problem of measuring round-trip delays. An advantage of this scheme is that it is similar to the host-loopback, which is a type of connectivity check using loopback OAM cells, and so should be easily implementable in systems which support the I.610 standard.

This scheme can be modified such that it finds the bottleneck nodes, by including in each cell a delay threshold value. Loopback at a node is performed

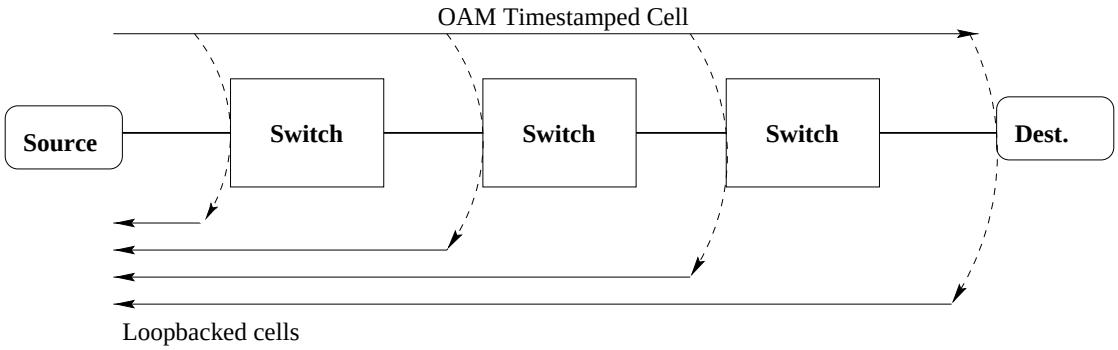

Figure 7.1: Back - Flooding of OAM performance management cells.

only if the queueing delay at that node is more than the threshold, or the node is unable to determine the per connection delays. The looped back cell may carry the address of the bottleneck switch for identification. This scheme, termed as selective back-flooding, is an improved version since a smaller bandwidth overhead is associated with it.

7.2.2 Queue jumping

A high priority cell that jumps to the head of its VC-queue upon arrival can be introduced in a connection. Such a cell can effectively measure the round trip delay along a connection. If a similar high priority cell sent later jumps all queues except one, the difference in the two measurements gives the delay in the queue the second cell did not jump. The address of the switch where the queue is not to be jumped can be written into the cell. If the cell jumps queues only in the forward path, the one way queueing delay can be accurately measured as well. This scheme has the advantages of being able to accurately measure the one way delay, and the delay at a particular switch. It is also practical as implementing a high priority cell in switches that employ per connection queueing is not difficult. Queues are usually handled as linked lists in software and a high priority cell can

be queued at the head of the list instead of the tail.

7.2.3 Alarm based schemes

Another method to pinpoint a bottleneck is by using soft-alarms. The number of cells in real-time traffic queues can be measured and alarms (called soft-alarms because they indicate soft-faults) can be sent back to the source of the VC when the count exceeds a threshold (which can be agreed upon at connection setup). This is scheme is different because information is pushed from the network elements to the measurement devices as compared to the other schemes where the information is gathered by the PM device. The bandwidth overhead here would be much smaller than in all the schemes described above.

7.3 Communication between PM Device and Switches

As evaluated in chapter 3, the distributed mechanism of exchange of information from the PM devices to the switches is preferable over the centralized method. It is also observable from the identification schemes proposed above that the bandwidth overhead of all the schemes is large. Moreover, identification would be required after a violation in QoS is found. When the PM device identifies the bottleneck switch it sends a message addressed to the switch to take corrective actions. This involves a delay of nearly 2.5 round trip times. This delay between actual congestion and the corrective actions is too large and can cause further problems.

In the distributed mechanism the PM device periodically inserts special cells carrying the measured and required delay. If a control action is required, the first

switch on the path that can afford to lower the local delay bound takes the corrective action. The switch also stamps the correction term on the forwarded special cell to pass the information to downstream nodes. In this way, if a corrective action is required, all switches in the path would contribute as much as they can afford. Similarly, if the measurements are much better than the contracted QoS, the switches on the path that are experiencing congestion may seize the opportunity to increase the local delay bound of the connection. In this scheme, the delay in control action after the measurement is performed is minimal, approximately one round trip time in the usual case.

This proposed heuristic algorithm requires further study. The format of the cell to communicate the information needs to be standardized. The algorithm for updating the connection parameters in various switches need to be formulated and evaluated.

Chapter 8

Conclusions

8.1 Summary

This research proposes a new performance management architecture which guarantees end-to-end QoS for real-time connections. The scheduling at one switch is an important constituent of this framework. The proposed hierarchical scheduler is capable of providing bandwidth to connections, and bounded delay or jitter guarantees to real-time connections independent of their bandwidth requirement. It is based upon the definition of urgency of a cell which increases with time in different ways for different types of connections. For the typical case of maximum delay sensitive connections, a new delay bounding scheduler (called HRDF) is proposed and analyzed in the context of leaky bucket bounded flows. The analysis determines the schedulability region of the HRDF scheduler, which is calculated to be less than the region of GPS but the same as the usable region of GPS. A review of the state-of-the-art of the technology used for scheduling reveals that the scheduler is implementable using the current hardware technology. To better understand the practical issues associated with an implementation of a scheduling discipline, the HRDF scheduler is implemented on a prototype commercial ATM switch manufactured by Lucent Technologies.

In order to guarantee end-to-end performance the delay and jitter statistics are measured on a per connection basis using in-service monitoring mechanisms. These measurements are performed with performance monitoring cells injected in the user cell stream by the PM device. Schemes for accurate measurement of delay and jitter using pattern cells and other specialized mechanisms are also developed in the thesis. Messages are sent to all nodes by the PM device if QoS violations or deteriorating patterns are detected. Periodic updates of QoS are also sent in order to facilitate switches to change parameters for better resource utilization. For QoS sensitive connections, the parameters controlling the urgency can be changed based upon end-to-end QoS measurements. A simple heuristic protocol is suggested in the thesis for this purpose.

8.2 Future Directions

A complete understanding of the traffic engineering framework requires answers to more questions. The problem of communicating the measurements to the switches in the nodes, and initiating corrective measures in the schedulers of the switches is an open problem. The heuristic presented in the thesis is only a first step. The effect of control actions of one connection on the network performance and on the performance of other connections (including a study of stability of the network) is also a topic for further research. In addition to the updates algorithm, the choice of the scheduling parameters of connections at connection setup also needs to be determined.

In the study of the hierarchical scheduler, the optimal scheduler for the rate shaped connections and the corresponding definition of urgency is also a topic for future research. The notion of urgency also needs to be defined for any other

scheduler that can be used as a delay or jitter bounding scheduler.

Finally, in order to understand the practical aspects of this framework, it needs to be implemented in a prototype switch. This work included an implementation and feasibility study of the HRDF scheduler. It needs to be extended to include QoS monitoring and signaling in order to make a functionally complete test-bed.

Appendix A

HRDF: The $N = 2$ Case

A.1 Fluid Analysis for $C_i = \infty$

Lemma A.1.1 For $N = 2$, and $x_1 \geq x_2$, the worst case delay for connection 2 is attained when both connections are greedy at the start of a system busy period ($t = 0$):

$$D_2^* = \frac{\sigma_1 + \sigma_1}{C - \rho_1(1 - \frac{x_2}{x_1})}$$

and worst case delay for connection 1 is attained when connection 2 is greedy at the start of system busy period ($t = 0$) and connection 1 is greedy at $t = \frac{\sigma_1}{C}(1 - \frac{x_2}{x_1})$:

$$D_1^* = \frac{\sigma_1 x_1 + \sigma_2 x_2}{C x_1}$$

whenever the condition

$$\frac{\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2} \geq \frac{\rho_1}{C} \tag{A.1}$$

is satisfied. If the condition is not satisfied, the worst case for connection 1 is also when both connections are greedy at time 0 and

$$D_1^* = \frac{\sigma_1 + \sigma_1}{C - \rho_1(1 - \frac{x_2}{x_1})} \frac{x_2}{x_1}$$

Proof: Consider a single connection. It is apparent that the maximum urgency will be achieved when the connection is greedy. The curve of urgency with time will be a triangle with highest point achieved when the last cell belonging to the burst (of size σ_i) is serviced. In the case of two connections, consider the interaction of two separate triangles of the urgency vs time curve. Specifically consider the rising edge (with slope x) of the connection with smaller slope (say connection 2, i.e., $x_1 \geq x_2$). Let the start of burst of connection 2 be at time 0 and that of connection 1 at time t_1. It is evident that the highest urgency achieved in the case of $t_1 \leq 0$ is when $t_1 = 0$. For $t_1 \geq 0$, the highest urgency is achieved when the urgency of connection 1 intersects that of connection 2 at its peak point. Thus these are the two cases of interest and the in-depth analysis of these cases follows.

Analysis of case 1: Consider the case for both sessions greedy at time 0, with $x_1 \geq x_2$. Thus, $\hat{A}_1(0, 0^+) = \sigma_1$ and $\hat{A}_2(0, 0^+) = \sigma_2$. Then till $t = \hat{t}^1$, the burst σ_1 of connection 1 is served at the capacity rate

$$\hat{t}^1 = \frac{\sigma_1}{C}. \tag{A.2}$$

For $\hat{t}^2 \geq t \geq \hat{t}^1$, connection 2 is served till the urgency of connection 1 catches up. Thus,

$$(\hat{t}^2 - \frac{\hat{t}^2 C - \sigma_1}{\rho_1}) x_1 = \hat{t}^2 x_2$$

$$\hat{t}^2 = \frac{\sigma_1}{C - \rho_1(1 - \frac{x_2}{x_1})}. \tag{A.3}$$

Now, the service will be provided to the connections in the ratio of r_1 to $1-r_1$ till the burst σ_2 is exhausted at $\hat{t}^3$

$$(t - \hat{t}^2 + (\hat{t}^2 - \frac{\hat{t}^2 C - \sigma_1}{\rho_1}) - \frac{(t - \hat{t}^2) r_1 C}{\rho_1}) x_0 = t x_1$$

$$r_1 = \frac{\rho_1}{C}(1 - \frac{x_2}{x_1}). \tag{A.4}$$

And,

$$(\hat{t}^3 - \hat{t}^2)(1 - r_1)C = \sigma_2$$

$$\begin{aligned} \hat{t}^3 &= \frac{\sigma_2}{(1 - r_1)C} + \hat{t}^2 \\ &= \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \end{aligned} \tag{A.5}$$

For $\hat{t}^3 \geq t \geq \hat{t}^4$, service is provided to both in the ratio s_1 to $1 - s_1$ till the queues clear out completely:

$$\hat{t}^4 = \frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2} \tag{A.6}$$

The maximum urgency attained in this case is

$$\begin{aligned} \hat{U}^* &= \hat{t}^3 x_2 \\ &= \frac{\sigma_1 x_2 + \sigma_2 x_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \end{aligned} \tag{A.7}$$

The arrival, departure and urgency profiles of both the connections is given in figure A.1 which also shows the various times defined above.

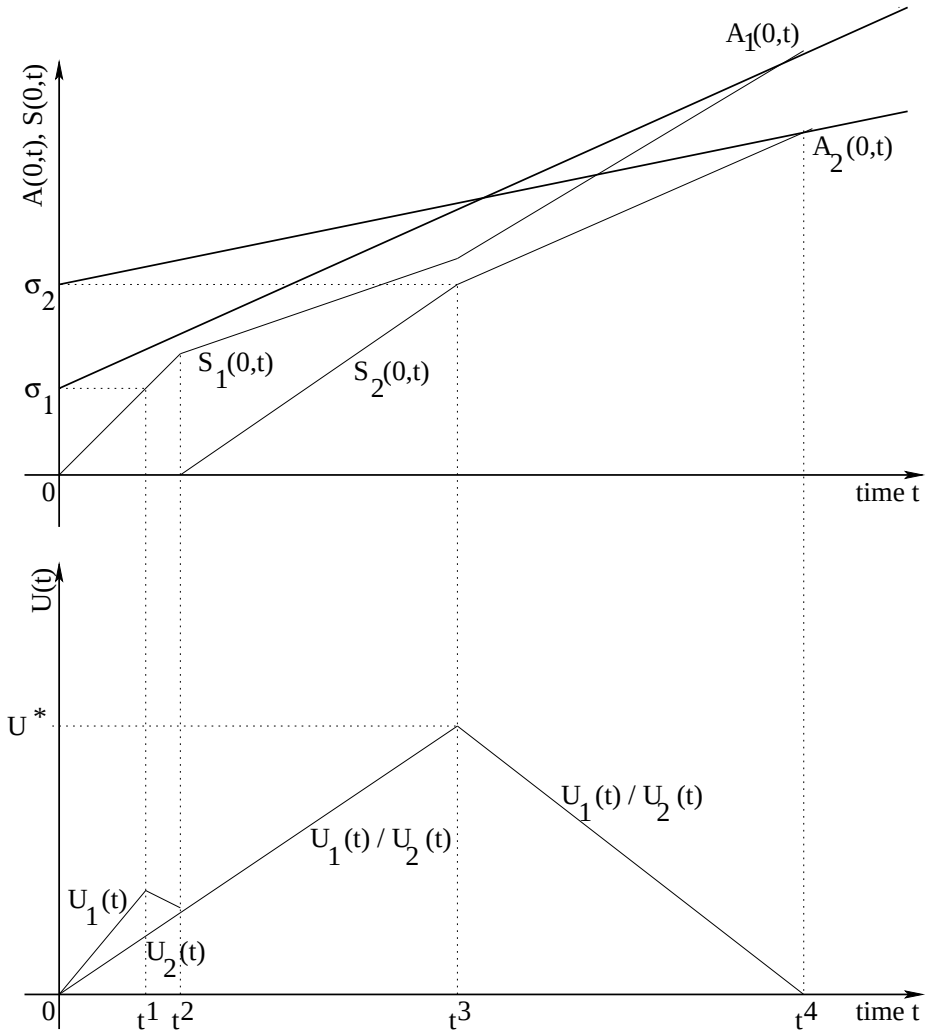

Figure A.1: Analysis for both sessions greedy at time 0.

Analysis of case 2: Consider the case for sessions 2 greedy at time 0 and session 1 greedy at time $\tilde{t}^1$,

$$\tilde{t}^1 = \frac{\sigma_2}{C}(1 - \frac{x_2}{x_1}). \tag{A.8}$$

In this case, the session 2 is served till time $\tilde{t}^2$,

$$\tilde{t}^2 = \frac{\sigma_2}{C}. \tag{A.9}$$

At time $\tilde{t}^2$, the urgency of both connections is equal. Note that this is the maximum possible urgency that connection 2 could attain. The burst from connection 1 is now served till time $\tilde{t}^3$,

$$\tilde{t}^3 = \frac{\sigma_1 + \sigma_2}{C}. \tag{A.10}$$

After $\tilde{t}^3$, the urgency of connection 1 falls, but is still higher till time $\tilde{t}^4$. Both connections are served then for $\tilde{t}^4 \geq t \geq \tilde{t}^5$ till the queues become empty at time $\tilde{t}^5$.

$$\tilde{t}^4 = \frac{\sigma_1 + \sigma_2 - \frac{\rho_1}{C}\sigma_2(1 - \frac{x_2}{x_1})}{C - \rho_1(1 - \frac{x_2}{x_1})}. \tag{A.11}$$

Figure A.2 illustrates this case pictorially. The maximum urgency attained in this case by connection 1 is

$$\tilde{U}^* = \frac{\sigma_1 x_1 + \sigma_2 x_2}{C}. \tag{A.12}$$

Thus,

$$\begin{aligned} \hat{D}_2 &= \frac{\hat{U}^*}{x_2} \\ &= \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \end{aligned} \tag{A.13}$$

$$\begin{aligned} \hat{D}_1 &= \frac{\hat{U}^*}{x_1} \\ &= \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})}\frac{x_2}{x_1} \end{aligned} \tag{A.14}$$

$$\begin{aligned} \tilde{D}_2 &= \tilde{t}^4 \\ &= \frac{\sigma_1 + \sigma_2 - \frac{\sigma_1}{C}\sigma_2(1 - \frac{x_2}{x_1})}{C - \rho_1(1 - \frac{x_2}{x_1})} \\ &\leq \hat{D}_2 \end{aligned} \tag{A.15}$$

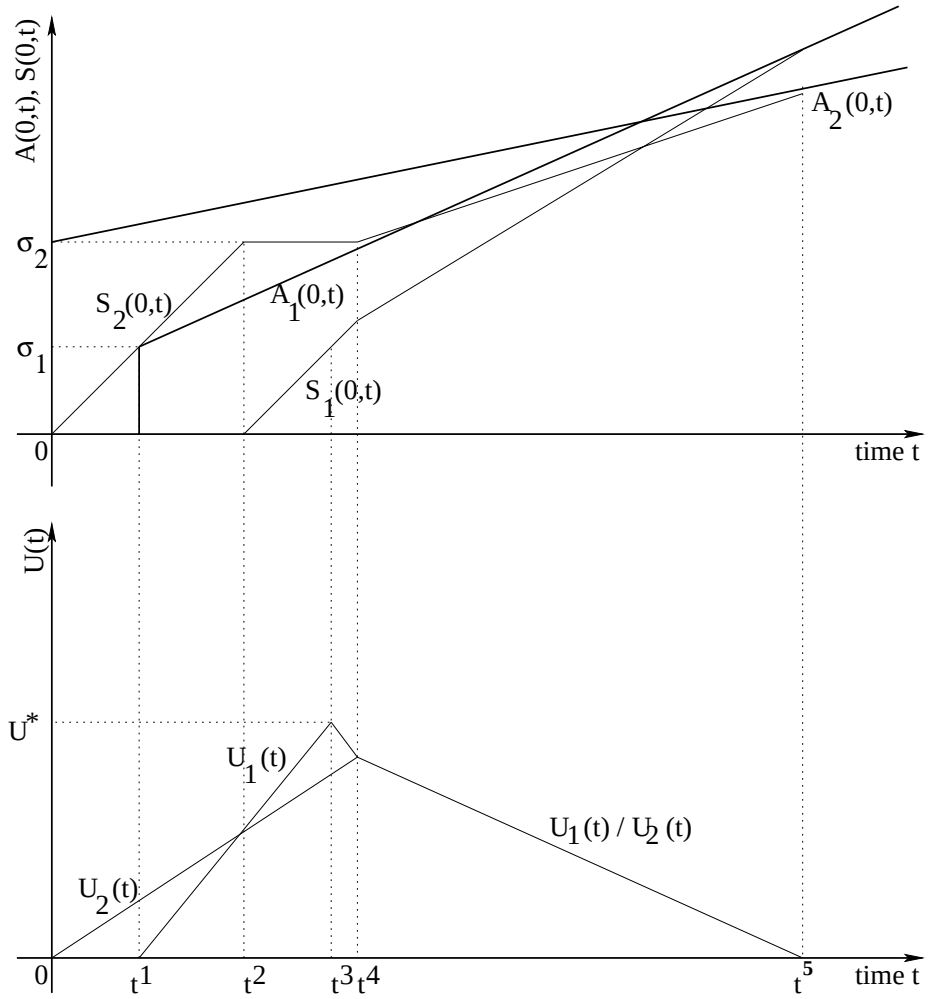

Figure A.2: Analysis of 2 greedy at time 0 and 1 greedy at $\tilde{t}^2$.

$$\begin{aligned} \tilde{D}_1 &= \frac{\tilde{U}^*}{x_1} \\ &= \frac{\sigma_1 x_1 + \sigma_2 x_2}{C x_1} \end{aligned} \tag{A.16}$$

Observe that under the condition in Eqn. (A.1),

$$\begin{aligned} \tilde{U}^* &\geq \hat{U}^* \\ \tilde{D}_1 &\geq \hat{D}_1 \end{aligned}$$

and thus,

$$D_1^* = \tilde{D}_1 = \frac{\sigma_1 x_1 + \sigma_2 x_2}{C x_1} \quad \text{(A.17)}$$

$$D_2^* = \hat{D}_2 = \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \quad \text{(A.18)}$$

As expected,

$$D_1^* \leq D_2^*.$$

If Eqn. (A.1) is not satisfied,

$$\begin{aligned} \tilde{U}^* &\leq \hat{U}^* \\ \tilde{D}_1 &\leq \hat{D}_1 \end{aligned}$$

and thus,

$$\begin{aligned} D_1^* &= \hat{D}_1 = \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \frac{x_2}{x_1} \\ D_2^* &= \hat{D}_2 = \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \end{aligned} \quad \text{(A.19)}$$

As expected,

$$D_1^* \leq D_2^*.$$

□

A.2 Comparison of connection delays with GPS

Lemma A.2.1 Consider a GPS scheduler with two connections ($x_1 \geq x_2$) assigned weights such that the worst case delay of one connection is the same as the worst

case delay of the HRDF scheduler. Then the worst case delay of the other connection is less in the Delay Scheduler compared to GPS under the condition in Eqn. (A.1).

Proof: Let ϕ_1 be the weight of connection 1 such that the worst case delay of connection 1 is $G_1^* = D_1^*$. The weight of connection 2 is then $1 - \phi_1$. The worst case delay for a GPS scheduler is attained when both connections are greedy at time 0 [11]. Thus,

$$C\phi_1 G_1^* = \sigma_1$$

or,

$$\phi_1 = \frac{\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2} \tag{A.20}$$

Note that under A.1,

$$\phi_1 \geq \frac{\rho_1}{C}$$

and also,

$$\begin{aligned} C(1-\phi_1)G_1^* &= \sigma_2 \frac{x_2}{x_1} \\ &\leq \sigma_2 \end{aligned}$$

Now let t_1^{clear} denote the time at which the queue for connection 1 is cleared. Thus,

$$\begin{aligned} t_1^{clear} &= \frac{\sigma_1}{C\phi_1 - \rho_1} \\ &= \frac{\sigma_1}{\frac{C\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2} - \rho_1} \end{aligned} \tag{A.21}$$

If $C(1-\phi_1)t_1^{clear} \geq \sigma_2$, i.e.,

$$\begin{aligned} \frac{\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2}(1 - \frac{x_2}{x_1}) &\leq \frac{\rho_1}{C} \\ \phi_1(1 - \frac{x_2}{x_1}) &\leq \frac{\rho_1}{C} \end{aligned} \tag{A.22}$$

then

$$\begin{aligned} G_2^* &= \frac{\sigma_2}{C(1-\phi_1)} \\ &= \frac{\sigma_1 x_1 + \sigma_2 x_2}{C x_2} \qquad (A.23) \\ &\geq \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \ \ under \ \ A.1 \\ &\geq D_2^* \qquad (A.24) \end{aligned}$$

else if $C(1-\phi_1)t_1^{clear} \leq \sigma_2$,

$$C(1-\phi_1)t_1^{clear} + (c - \rho_1)(G_2^* - t_1^{clear}) = \sigma_2$$

i.e.,

$$\begin{aligned} G_2^* &= \frac{\sigma_1 + \sigma_2}{C - \rho_1} \\ &\geq \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \\ &\geq D_2^* \qquad (A.25) \end{aligned}$$

Thus under both the conditions, the worst case delay of connection 2, G_2^*, is greater than the worst case delay of the Delay scheduler, D_2^*. Now consider the reverse case, $G_2^* = D_2^*$. If,

$$C(1-\phi_2)G_2^* \leq \sigma_1 + \rho_1 G_2^* \tag{A.26}$$

then,

$$C\phi_2 G_2^* = \sigma_2$$

i.e.,

$$\phi_2 = (1 - \frac{\rho_1}{C}(1 - \frac{x_2}{x_1}))\frac{\sigma_2}{\sigma_1 + \sigma_2} \tag{A.27}$$

Note that with ϕ_2 defined as above A.26 is always satisfied. Hence,

$$\begin{aligned} G_1^* &= \frac{\sigma_1}{C(1-\phi_2)} \\ &= \frac{\sigma_1 + \sigma_2}{C + \frac{\sigma_2}{\sigma_1}\rho_1(1 - \frac{x_2}{x_1})} \\ &\leq G_2^* \end{aligned} \tag{A.28}$$

Observe that under A.1,

$$G_1^* \geq D_1^* \tag{A.29}$$

□

Lemma A.2.2 Consider a GPS scheduler with two connections ($x_1 \geq x_2$) assigned weights such that the worst case delay of one connection is the same as the worst case delay of the HRDF scheduler. Then the worst case delay of the other connection is less in the Delay Scheduler compared to GPS when the condition A.1 is not satisfied.

Proof: In this case, lemma A.1.1 gives that

$$\begin{aligned} D_1^* &= \hat{D}_1 = \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})} \frac{x_2}{x_1} \\ D_2^* &= \hat{D}_2 = \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1})}. \end{aligned} \tag{A.30}$$

$G_2^* = D_2^*$. From the previous lemma, $\phi_1 \geq \frac{\rho_1}{C}$ under the condition of Eqn. (A.1). Thus $\phi_1 \leq \frac{\rho_1}{C}$, and the worst case delay of 2 will be achieved first. The worst case delay of 1 will be achieved after the queue of connection 2 becomes empty and the fraction for 2 increases. Therefore,

$$\begin{aligned} C\phi_2 G_2^* &= \sigma_2 \\ \phi_2 &= \frac{\sigma_2}{CD_2^*} \\ 1 - \phi_2 &= \frac{CD_2^* - \sigma_2}{CD_2^*} \\ &= \frac{C\sigma_1 + \rho_1\sigma_2(1 - \frac{x_2}{x_1})}{C\sigma_1 + C\sigma_2}. \end{aligned} \tag{A.31}$$

It can be checked that $1 - \phi_2 \leq \frac{\rho_1}{C}$ whenever $\frac{\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2} \leq \frac{\rho_1}{C}$. The worst case of connection 1 is achieved at time t_0:

$$t_0 = \frac{\sigma_2}{C\phi_2 - \rho_2}. \tag{A.32}$$

Now note that

$$C(1 - \phi_2)t_0 \geq \sigma_1$$

Therefore,

$$G_1^* = t_0 - \frac{C(1 - \phi_2)t_0 - \sigma_1}{\rho_1}$$

$$
\begin{aligned}
&= \frac{\sigma_1+\sigma_2}{\rho_1} + t(1-\frac{C-\rho_2}{\rho_1}) \\
&= \frac{\sigma_1+\sigma_2}{\rho_1} + \frac{D_2^*\sigma_2}{\sigma_2-\rho_2 D_2^*}(1-\frac{C-\rho_2}{\rho_1}) \\
&= \frac{D_2^*}{\rho_1}\{C-\rho_1(1-\frac{x_2}{x_1}) + \frac{D_2^*\sigma_2}{\sigma_2-\rho_2 D_2^*}(\rho_1+\rho_2-C)\} \\
&= \frac{D_2^*}{\rho_1}\ \frac{\sigma_2\rho_1 x_2 - \sigma_1\rho_2 x_1}{x_1(\sigma_2-\rho_2 D_2^*)}
\end{aligned}
$$

Observe that when $\frac{\sigma_1 x_1}{\sigma_1 x_1+\sigma_2 x_2} \leq \frac{\rho_1}{C}$,

$$\frac{\sigma_2\rho_1 x_2 - \sigma_1\rho_2 x_1}{\rho_1 x_1(\sigma_2-\rho_2 D_2^*)} \geq \frac{x_2}{x_1}$$

And thus,

$$G_1^* \geq D_1^*. \tag{A.33}$$

Hence the worst case delay of GPS for connection 1 is more than the worst case delay of connection 1 in HRDF scheduler. Now consider the case of $G_1^* = D_1^*$. If connection 2 clears its queue before connection 1 clears its burst of σ_1 at time t_1:

$$
\begin{aligned}
C\phi_1 t_1 + (C-\rho_2)(G_1^* - t_1) &= \sigma_1 \\
C(1-\phi_1)t_1 &= \sigma_2 + \rho_2 t_1
\end{aligned}
$$

Also,

$$
\begin{aligned}
(C-\rho_2)D_1^* &= \sigma_1+\sigma_2 \\
or,\quad \frac{x_2}{x_1} &= \frac{C-\rho_1}{C-\rho_1-\rho_2}
\end{aligned}
$$

which is not possible since $\frac{x_2}{x_1} \leq 1$ and $\frac{C-\rho_1}{C-\rho_1-\rho_2} \geq 1$. Connection 1 therefore clears its burst before the queue for connection 2 clears out at time t_1. So,

$$
\begin{aligned}
C\phi_1 t_1 &= \sigma_1 + \rho_1(t_1 - G_1^*) \\
C(1-\phi_1)t_1 &= \sigma_2 + \rho_2 t_1 \\
or \quad t_1 &= \frac{\sigma_1 + \sigma_2 - \rho_1 D_1^*}{C - \rho_1 - \rho_2}
\end{aligned}
$$

Thus the maximum delay for connection 2 is:

$$
\begin{aligned}
G_2^* &= \frac{\sigma_2}{C(1-\phi_1)} \\
&= \frac{\sigma_2 t_1}{\sigma_2 + \rho_2 t_1} \\
&= \frac{\sigma_2(\sigma_1 + \sigma_2 - \rho_1 D_1^*)}{C\sigma_2 + \rho_2\sigma_1 - \rho_1\sigma_2 - \rho_1\rho_2 D_1^*} \\
&= \frac{\sigma_2(\sigma_1 + \sigma_2)(C - \rho_1)}{(C-\rho_1)(C\sigma_2 + \rho_2\sigma_1 - \rho_1\sigma_2) + (C\rho_1\sigma_2 - \rho_1^2\sigma_2 - \rho_1\rho_2\sigma_2)\frac{x_2}{x_1}} \\
&= \frac{\sigma_1 + \sigma_2}{C - \rho_1(1 - \frac{x_2}{x_1}) + \rho_2(\frac{\sigma_1}{\sigma_2} - \frac{\rho_1}{C-\rho_1}\frac{x_2}{x_1})}. \qquad \text{(A.34)}
\end{aligned}
$$

Consequently, under $\frac{\sigma_1 x_1}{\sigma_1 x_1 + \sigma_2 x_2} \leq \frac{\rho_1}{C}$,

$$G_2^* \geq D_2^*.$$

□

BIBLIOGRAPHY

[1] ATM Forum, *Traffic Management Specification Version 4.1*, Mar 1999. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121.000.pdf.

[2] ITU-T Recommendation I.356, *B-ISDN ATM layer cell transfer performance*, Mar 2000. http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-I.356-200003-I.

[3] ITU-T Recommendation I.371, *Traffic Control and Congestion Control in B-ISDN*, Mar 2000. http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-I.371-200003-I.

[4] ITU-T Recommendation I.610, *B-ISDN Operations and Maintenance Principles and Functions*, Feb 1999. http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-I.610-199902-I.

[5] D. E. McDysan, *ATM: Theory and Practice*. McGraw-Hill, 1995. http://shop.mcgraw-hill.com/cgi-bin/pbg/0070453462.html?id=86Lc4fcH.

[6] N. Giroux and S. Ganti, *Quality of Service in ATM Networks: State-of-the-art traffic managment*. Prentice Hall PTR, 1998. http://vig.prenhall.com/catalog/professional/product/1,4096,0130953873,00.html.

[7] S. Keshav, *An Engineering Approach to Computer Networking*. Addison Wesley Longman, Inc., 1997. http://cseng.aw.com/book/0,,0201634422,00.html.

[8] B. P. Nick McKeown and M. Zhu, "Matching output queueing with combined input and output queueing," in *Proceedings of the 35th Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, Illinois), October 1997. http://tiny-tera.stanford.edu/~nickm/papers/Allerton97.pdf.

[9] N. McKeown, "Publications's web-page." http://tiny-tera.stanford.edu/~nickm/papers/index.html.

[10] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of fair queueing algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 19, pp. 1–12, September 1989.
http://netweb.usc.edu/cs551/papers/Demers.pdf.

[11] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, June 1993.
http://doi.acm.org/10.1145/159907.159914.

[12] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 137–150, April 1994.
http://doi.acm.org/10.1145/187037.187047.

[13] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *Proceedings of Infocom*, (Toronto), pp. 636–646, June 1994.

[14] J. C. R. Bennet and H. Zhang, "Hierarchical packet fair queueing algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 675–689, October 1997.
http://doi.acm.org/10.1145/268715.268724.

[15] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375–385, June 1996.
http://doi.acm.org/10.1145/230719.230732.

[16] N. Matsufuru and R. Aibara, "Efficient fair queueing for ATM networks using uniform round robin," *Proceedings of Infocom*, March 1999.
http://www.ieee-infocom.org/1999/papers/03b_02.pdf.

[17] L. Georgiadis, R. Guerin, and A. K. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," in *Proceedings of Infocom*, pp. 524–532, 1994.
http://pender.ee.upenn.edu/~guerin/publications/edf.ps.gz.

[18] D. E. Wrege and J. Liebeherr, "A near-optimal packet scheduler for QoS networks," in *Proceedings of IEEE '97*, (Kobe, Japan), pp. 576–583, 1997.
ftp://ftp.cs.virginia.edu/pub/jorg/papers/monotonic.pdf.

[19] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368–379, 1990.
http://www.research.ibm.com/people/d/dverma/papers/jsac1990.pdf.

[20] D. C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proceedings of Tricomm 91*, (Chapel Hill, North Carolina), pp. 35–46, April 1991.
http://www.research.ibm.com/people/d/dverma/papers/Tricomm1991.pdf.

[21] D. McDonald, R. Liao, and G. N., "Variation fluctuation smoothing for ATM circuit emulation service," *Proceedings of ITC*, vol. 15, pp. 761–770, June 1997.
http://comet.columbia.edu/~liao/publications/vfs.pdf.

[22] S.-K. Kweon and K. G. Shin, "Providing deterministic delay guarantees in ATM networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 838–850, December 1998.
http://doi.acm.org/10.1145/300386.300412.

[23] M. Andrews and L. Zhang, "Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule," in *Proceedings of Infocom*, (New York), March 1999.
http://www.ieee-infocom.org/1999/papers/03b_01.pdf.

[24] D. A. Hayes, M. Rumsewicz, and L. L. H. Andrew, "Quality of service driven packet scheduling disciplines for real time applications: Looking beyond fairness," in *Proceedings of Infocom*, (New York), March 1999.
http://www.ieee-infocom.org/1999/papers/03b_04.pdf.

[25] T. M. Chen *et al.*, "Monitoring and control of ATM networks using special cells," *IEEE Network*, pp. 28–38, Sept/Oct 1996.
http://www.komunikasi.org/acrobat/atm/Monitoring_and_Control_of_ATM_using_Special_Cells.pdf.

[26] T. M. Chen and S. S. Liu, "Monitoring and control of strategic and tactical networks using control packets," in *Proceedings ATIRP Consortium*, pp. 183–187, Feb 1999.

[27] C. Roppel, "Estimating cell transfer delay in ATM networks using in-service monitoring," in *Proceedings Globecom*, pp. 904–908, 1995.

[28] S. Banerjee, D. Tipper, and B. H. M. Weiss, "Traffic experiments on the vBNS wide area ATM network," *IEEE Communications Magazine*, pp. 126–133, Aug 1997.
http://www2.sis.pitt.edu/~mweiss/papers/tipper.pdf.

[29] D. Gaïti and G. Pujolle, "Performance management issues in ATM networks: Traffic and congestion control," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 249–257, Apr 1996.
http://doi.acm.org/10.1145/225989.226017.

[30] T. Lindh, "Performance management in switched ATM networks," in *Proceedings, Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services*, pp. 439–450, March 1998.

[31] R. G. Cheng, C. J. Chang, and L. F. Lin, "A QoS-provisioning neural fuzzy connection admission controller for high speed networks," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 111–121, Feb 1999.
http://doi.acm.org/10.1145/299905.299915.

[32] M. Siler and J. Walrand, "On-line measurement of QoS for call admission control," in *Proceedings, Sixth IEEE/IFIP International Workshop on QOS*, (Napa), pp. 39–48, May 1998.
http://walrandpc.eecs.berkeley.edu/Papers/qos98.pdf.

[33] B. Bensaou, S. T. C. Lam, H.-W. Chu, and D. H. K. Tsang, "Estimation of the cell loss ratio in ATM networks with a fuzzy system and application to measurement-based admission control," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 572–584, Aug 1997.
http://doi.acm.org/10.1145/262028.262041.

[34] RAD Data Communications - White Paper, *Customer Premises ATM NTUs: Cost Effective End-to-End Management of ATM Services.*
http://www.rad.com/networks/whitepap.htm.

[35] T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "Delay differentiation and adaptation in corestateless networks," in *Proceedings of IEEE INFOCOM*, (Tel Aviv, Israel), March 2000.
http://www.ieee-infocom.org/2000/papers/651.ps.

[36] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications.* Wiley Interscience, New York, 1976.
http://www.wiley.com/Corporate/Website/Objects/Products/0,9049,89913,00.html.

[37] L. Kleinrock, "A delay dependent queue discipline," *Naval Research Logistics Quarterly*, vol. 11, pp. 329–341, December 1964.

[38] J. R. Jackson, "Some problems in queueing with dynamic priorities," *Naval Research Logistics Quarterly*, vol. 7, pp. 235–247, September 1960.

[39] J. R. Jackson, "Waiting time distributions for queues with dynamic priorities," *Naval Research Logistics Quarterly*, vol. 19, pp. 31–36, March 1962.

[40] F. M. Chiussi and A. Francini, "Implementing fair queueing in atm switches: The discrete-rate approach," in *Proceedings of IEEE Infocom*, (San Francisco, CA), pp. 272–281, April 1998.
http://www.ieee-infocom.org/1998/papers/03a_1.pdf.

[41] Marconi Systems, *ATM Switch Network Modules Product Overview.* http://www.marconi.com/html/solutions/atmswitchnetworkmodules% productoverview.htm.

[42] Cisco Systems, *Data Sheets for MGX family of Edge Concentrators.* http://www.cisco.com/warp/public/cc/pd/si/mg8200/mg8250/prodlit/ index.shtml.

[43] Cisco Systems, *QoS on Catalyst 6000 Family Switches: Output Scheduling on the Catalyst 6000 with PFC Using Hybrid Mode.* http://www.cisco.com/warp/public/473/60.html.

[44] Cisco Systems, *LightStream 2020 System Overview.* http://www.cisco.com/univercd/cc/td/doc/product/atm/l2020/2020r231/ sysover/index.htm.

[45] Cisco Systems, *Per-VC Class-Based, Weighted Fair Queuing (Per-VC CB-WFQ) on the Cisco 7200, 3600, and 2600 Routers.* http://www.cisco.com/warp/public/121/7200_per-vc-CBWFQ.html.

[46] Cisco Systems, *Guide to the ATM Technology.* http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/ pereg_1/atm_tech.

[47] Alcatel, *The OMNI family QOS framework.* http://www.cid.alcatel.com/industry_analysts/pdf/omni_qos.pdf.

[48] Alcatel, *Alcatel 7420: Edge Services Router.* http://www.cid.alcatel.com/doctypes/product/html/a7420.jhtml.

[49] Lucent Technologies, *CBX 500 Multi Service WAN Switch: ATM I/O Modules.* http://www.lucent.com/livelink/154204_TechnicalMaterial.pdf.

[50] Lucent Technologies, *PacketStar[tm] PSAX 1250 Multiservice Media Gateway.* http://www.lucent.com/products/solution/0,,CTID+2007-STID+ 10075-SOID+944-LOCL+1,00.html.

[51] Lucent Technologies, *GX550: ATM Base I/O Modules.* http://www.lucent.com/livelink/139839_Brochure.pdf.

[52] Nortel Networks, *Nortel Networks ATM services on Passport multiservice platforms.* http://www.nortelnetworks.com/products/library/collateral/55141.02-11-00. pdf.

[53] Nortel Networks, *Nortel Networks Passport 15000 Multiservice Switch.* http://www.nortelnetworks.com/products/library/collateral/80015.02-11-00.pdf.

[54] General DataComm, *GDC Apex Packet Switches: Multi Service Packet Switching.* http://www.gdc.com/inotes/pdf/apexfam.pdf.

[55] General DataComm, *GDC's Multi-tier Shaping: Advanced ATM Traffic Management for New Broadband Applications.* http://www.gdc.com/inotes/pdf/mts%20paper.pdf.

[56] GlobeSpan Inc., *ATecoM ATM products.* http://www.globespan.net/products/product5.html.

[57] Conexant Systems Inc., *Conexant OC3 to OC48 Traffic Stream Processor Family Network Processors.* http://ebiz4.conexant.com/default.sph/SaServletEngine.class/Web/products/solutionsubcategories.jsp?SolnFamId=501&SolnCatId=526&SolnSubCatId=589.

[58] Transwitch Corp., *CUBIT-3: Multi-PHY CellBus Access Device.* http://www.transwitch.com/files/to/cub32_o_.pdf.

[59] Transwitch Corp., *ATM AccessEDGE User's Guide.* http://www.transwitch.com/files/ug/aedg1_g1.pdf.

[60] PMC-Sierra Inc., *PM7326 (S/UNI-APEX).* http://www.pmc-sierra.com/products/details/pm7326/.

[61] LSI Logic Corp., *Technical Manual: L64364 ATMizer II+ ATM-SAR Chip.* http://www.lsilogic.com/techlib/techdocs/networking/L64364.pdf.

[62] Acorn Networks, *genFlow ACN-2500gF.* http://www.acorn-networks.com/Products/genFlow_ACN-2500gF.html.

[63] ZettaCom Inc., *ZEN Multi-Protocol Processing Family.* http://www.zettacom.com/products/.

[64] Applied Micro Circuits Corp., *Product Releases: nPX5700 10 Gbps Traffic Manager.* http://www.mmcnetworks.com/PressRelations/pr_prod_40901.asp.

[65] Vitesse Semiconductor Corp., *PaceMaker Products.* http://www.vitesse.com/products/subcategories.cfm?family_id=6&category_id=18&subcategory_id=4.

[66] Vitesse Semiconductor Corp., *Vitesse Announces General Availability of the World's First OC-48 SAR and Traffic Management Chip - PaceMaker 2.4.* http://www.oro-logic.com/news/050801-1.shtml.

[67] Azanda Network Devices, *Technology: Traffic Management.* http://www.azanda.com/prod.htm.

[68] Silicon Access Networks, *Terabit Router ChipSet Products.* http://www.siliconaccess.com/products/.

[69] Bay Microsystems Inc., *Solutions.* http://www.baymicrosystems.com/solutions/index.html.

[70] A. Arora and J. S. Baras, "Performance monitoring in ATM networks," tech. rep., CSHCN and ISR, University of Maryland, April 1998. http://www.isr.umd.edu/TechReports/CSHCN/1998/CSHCN_TR_98-12/CSHCN_TR_98-12.phtml.

[71] A. Arora, J. S. Baras, and G. Mykoniatis, "Delay monitoring in ATM networks," in *Proceedings ATIRP Consortium*, pp. 259–263, Feb 1999.