Simulation of Lead-Time Scheduling in PMP FWA Networks

Robert Buchnajzer

A Thesis

In

The Department

Of

Computer Science

Presented in Partial Fulfillment of the Requirements for the Degree of
Master of Computer Science at Concordia University
Montreal, Quebec, Canada

January 2003

0-612-82655-4

Abstract

Simulation of Lead-Time Scheduling in PMP FWA Networks

Robert Buchnajzer

The terrestrial Point-to-Multipoint Fixed Wireless Access (PMP FWA) systems have evolved from their first circuit-switched proprietary implementations in the 1970s into flexible, packet-based solutions available to Competitive Local Exchange Carriers (CLEC) for providing POTS and internet access in WAN and MAN. Examples of such systems can be seen in a number of proprietary technologies, as well as in the commercial implementations of the IEEE 802 standards.

The purpose of this thesis is to evaluate the performance of a PMP FWA system in the context of today's packet based IntServ infrastructure. The evaluation takes into account three scheduling algorithms (FIFO, strict priority, and lead-time) in a number of different configurations. This thesis also proposes a set of mechanisms that can be introduced into the IntServ framework to support the lead-time scheduling algorithm.

Acknowledgements

I would like to thank both my thesis supervisors, Dr. J. W. Atwood and Dr. A. Krzyzak, for their involvement in this work. Their input to this work has been invaluable.

Table of Contents

List of Figures vii

List of Tables ix

1 Introduction 1

1.1 Real-Time Packet Traffic in the Context of a Packet-based Network 1

1.2 Real-Time Scheduling 3

1.3 Aims and Motivation of Thesis 5

2 Real-Time Traffic 6

3 QoS 9

3.1 DiffServ 10

3.2 IntServ 13

3.2.1 RSVP 17

3.3 Packet Scheduling 23

3.3.1 Strict Priority 26

3.3.2 Round-Robin 26

3.3.3 Deficit Round-Robin 27

3.3.4 Weighted Round-Robin 27

3.3.5 Fair Queuing/Virtual Clock 28

3.3.6 Weighed Fair Queuing 28

3.3.7 Delay Earliest Due Date 29

3.3.8 FIFO+ 29

3.3.9 Jitter Earliest Due Date 29

3.3.10 Lead-Time Scheduling 30

3.3.11 Other Non-Work Conserving Schedulers 30

3.4 RTP 31

3.4.1 RTP Data 32

3.4.2 RTP Control 33

3.5 Discussion 33

4 Deadline Distribution for Lead-Time Scheduling 35

5 Problem Statement ... 44
6 Support for Lead-Time Scheduling in IntServ ... 46
6.1 Support for Lead-Time Scheduling in IP ... 46
6.2 Support for Lead-Time Scheduling in RSVP ... 48
7 Simulation Environment ... 51
7.1 MAC Layer ... 56
7.2 Requirements for Voice Communication ... 57
7.3 Configuration of the Simulation System 1 ... 58
7.4 Configuration of the Simulation System 2 ... 62
7.5 Simulation Input and Results Generation ... 63
8 Simulation Results ... 65
8.1 Simulation System 1 ... 65
8.1.1 Packet Trip-Time ... 65
8.1.2 Queue Occupancy ... 67
8.2 Simulation System 2 ... 71
8.2.1 Packet Trip-Time ... 71
8.2.2 Queue Occupancy ... 72
9 Conclusion ... 75
9.1 Current Limitations and Future Work ... 75
Bibliography ... 76

List of Figures

Figure 1: IPv4 ToS Field ... 11
Figure 2: DiffServ Code Point ... 12
Figure 3: IntServ Architecture ... 14
Figure 4: RSVP Signaling ... 21
Figure 5: RSVP Message Format ... 22
Figure 6: RSVP Object Format ... 22
Figure 7: Simple Packet Scheduler ... 24
Figure 8: Example of a Queuing System ... 25
Figure 9: Plot of the Density U(A,B) of G_n ... 36
Figure 10: Plot of the cdf G_n ... 36
Figure 11: Plot of 1 - G_n ... 37
Figure 12: Regions of H_n ... 37
Figure 13: Plot of H_n ... 39
Figure 14: Plot of F_n ... 41
Figure 15: Regions of F_n ... 41
Figure 16: IPv4 Header ... 47
Figure 17: Simple PMP-FWA Network ... 53
Figure 18: PMP-FWA Network with a Repeater ... 55
Figure 19: PMP-FWA Network ... 56
Figure 20: PMP-FWA Simulation System ... 59
Figure 21: Simplified PMP-FWA Simulation System 1 ... 60
Figure 22: Simplified PMP-FWA Simulation System 2 ... 62
Figure 23: FIFO; Queue Occupancy for Node 1 ... 68
Figure 24: FIFO: Queue Occupancy for Node 2 ... 68
Figure 25: FIFO: Queue Occupancy for Node 3 ... 68
Figure 26: FIFO: Queue Occupancy for Node 4 ... 68
Figure 27: Strict-Priority: Queue Occupancy for Node 1 ... 69
Figure 28: Strict-Priority: Queue Occupancy for Node 2 ... 69
Figure 29: Strict-Priority: Queue-Occupancy for Node 3 ... 69

Figure 30: Strict-Priority: Queue Occupancy For Node 4 69
Figure 31: Lead-Time: Queue Occupancy for Node 1 70
Figure 32: Lead-Time: Queue Occupancy for Node 2 70
Figure 33: Lead-Time: Queue Occupancy for Node 3 70
Figure 34: Lead-Time: Queue Occupancy for Node 4 70
Figure 35: FIFO; Queue Occupancy for Node 1 73
Figure 36: FIFO: Queue Occupancy for Node 4 73
Figure 37: Strict-Priority: Queue Occupancy for Node 1 73
Figure 38: Strict-Priority; Queue Occupancy for Node 4 74
Figure 39: Lead-Time; Queue Occupancy for Node 1 74
Figure 40: Lead-Time; Queue Occupancy for Node 4 74

List of Tables

Table 1: Most Frequently Used RSVP Objects 23

Table 2: Link Capacity of the Simulation System 61

Table 3: FIFO Packet Trip Time Statistics 65

Table 4: Strict Priority Packet Trip Time Statistics 65

Table 5: Lead Time Packet Trip Time Statistics 66

Table 6: FIFO Packet Trip Time Statistics 71

Table 7: Strict Priority Packet Trip Time Statistics 71

Table 8: Lead Time Packet Trip Time Statistics 72

1 Introduction

1.1 *Real-Time Packet Traffic in the Context of a Packet-based Network*

With the discovery of the Internet by the average consumer, the demand for bandwidth has been growing exponentially during the past few years. The large volume of traffic going through the packet switched networks is overwhelming the carriers. The usage of the Internet has evolved from simple e-mail used by the pioneers of this new communication medium, to fully interactive multimedia applications rich in sound and video. Indeed, today's Internet carries traffic ranging across the whole spectrum from real-time voice and video approaching the quality of a DVD stream multiplexed with various types of traffic such as web browsing and file sharing.

The IP based networks were designed at the beginning for an efficient sharing of the links, a scarce and precious resource, that interconnect the routers. However, the IP protocol, although very well engineered for non-real-time traffic, does not stand up to the challenge in the context of today's data traffic imposed by the latest consumer trends. This is because the traditional legacy packet routing mechanisms are based on the best effort service [RFC791], which is well suited for applications with lax demands for timely and predictable packet delivery. In such a network, the protocols at the transport layer are responsible for timely packet delivery [RFC1889], very frequently discarding those which have been received after a short period of congestion, or delivering to the application data packets that have lost relevance in the temporal context of the data stream.

This shortcoming of the best effort mechanism has precipitated an introduction of the DiffServ [DiffServ] and IntServ [IntServ] frameworks for negotiating and providing a certain level of predictability from the network. These mechanisms discriminate different flows of traffic, and discriminate data traffic based on different end-to-end delivery characteristics. However, most of these mechanisms take into account the temporal relevance of the information across its lifetime on the network only to a limited extent. The QoS parameters requested by the application during the handshaking process, although they can be renegotiated in the middle of a connection, represent a firm contract at each node along the path of the stream. The network then, provides the requested QoS, not considering temporary surges in the load on one or more of the routers along the path of the stream. Consequently, during the short periods of congestion at one of the nodes, the traffic delayed at the busy routers might unnecessarily get discarded by the scheduling algorithms, degrading the perception of the quality of the stream. This is especially true in different types of audio traffic [KBSSGM98] [KT01], such as music, which are inherently extremely sensitive to the loss of data, and most of the time it shows itself as clicks or pauses in the stream.

The protocols for negotiating the QoS such as RSVP [RFC2205] of the IntServ [RFC2210] framework, allow the application to specify the timing requirements of the stream, including the allowed tolerance for the jitter, packet loss, and the packet trip time. However, these parameters are established at each hop without the a-priori knowledge about the actual amount of time that the packet is going to spend at each individual hop along the path of the stream. The scheduling algorithms, such as EDF, WFQ and different variants of Round-Robin [Zhang95] try to approximate the amount of time that a packet should spend in a local router. However, even a serious delay in one of the routers does not necessarily mean that the packet is not going to meet its QoS requirements. This is because the load on the other routers along the path might be much lighter, compensating for the temporary delay at only one of the nodes.

1.2 Real-Time Scheduling

Recent work [DLS00] [KLSY02] in queuing theory has introduced the concept of lead-time scheduling. In order to meet the timing requirements of each stream, the schedulers keep track of the customer lead-times, where the lead-time is defined as the time remaining until the deadline elapses, that is

$$leadTime = deadline - currentTime$$

Customer lead-time decreases linearly while the customer is in the queue. By keeping track of the total lead time at each of the nodes along the path of the stream, the network can gain a considerable elasticity during the congestion periods, while still being able to satisfy the stringent timing requirements of interactive real-time applications.

Doytchinov et al. [DLS00] analyze this scheduling discipline from the point of view of heavy traffic analysis (queuing system in which the traffic intensities at each node approach 1). This type of analysis studies the behavior of queue length and workloads rather than focusing on the lead-times of individual customers. Moreover, in their work the authors do not take into account the network behavior based on this algorithm.

Kruk et al. [KLSY02] take the work done in [DLS00] further by generalizing the results in [DSL00] to acyclic queuing networks. In their analysis, they determined the fraction of customers that exit the system late.

Kandlur et al. [KSF91] have proposed an algorithm for real-time scheduling which deals with the establishment of the channel, computing a guaranteed end-to-end delay for messages belonging to a channel, and the scheduling of packets to achieve this goal. In their algorithm they are using the deadlines of the delivery of the messages, along with the latency of the channel and the response time of the scheduler. Although the concept presented in this paper is very similar to Doytchinov et al. [DLS00], it presents the analysis of the problem from the feasibility and correctness point of view, not considering the performance in a queuing network. The authors use the Earliest Date Due (EDD) scheduling discipline, and show that this is an optimal scheduler for solving this problem. In their proposal, the packets are carrying the deadlines, and a network time coordination protocol (for example, NTP) coordinates the time base between the nodes.

A very similar mechanism designed for dealing specifically with jitter was presented by Clark et al. [CSZ92]. Their FIFO+ scheduling policy tracks the average queuing delay experienced by all packets at a particular node through a low-pass filter. On departure, the amount of time that a packet's individual delay differed from the average delay is added to a delay variance accumulated in the packet's header. Packets are served in the order of this difference timestamp. Clark et al. [CSZ92] have shown that FIFO+ reduces delay variance.

1.3 Aims and Motivation of Thesis

The objective of this thesis is to present the lead-time scheduling algorithm in the context of today's packet based infrastructure carrying real-time traffic. The first part of this thesis gives the current state of the processing of real-time traffic, and the technological advances that were accomplished during the last few years to accommodate this new type of payload on the existing infrastructure of the Internet.

The secondary objective of this thesis is to evaluate the performance of a lead-time scheduler in the context of a packet based access network for real-time voice. Although the evaluation presented in this thesis is not specific to an access network as such, the availability of the performance measurements for voice traffic makes this platform a compelling choice for evaluating this algorithm.

The presentation of the work in this thesis proceeds as follows: Chapter 2 gives an overview of the characteristics and requirements for real-time traffic. Chapter 3 describes the current mechanisms available for providing end-to-end QoS. Chapter 4 develops an aspect of the theory of the lead-time scheduling. Chapter 5 exposes the problem statement that this thesis is addressing. Chapter 6 presents the context in which the lead-time scheduler is evaluated. Chapter 7 evaluates the performance of the lead-time scheduler. Chapter 8 concludes the work with suggestions for future research that might be based on lead-time scheduling.

2 Real-Time Traffic

Traditional communication network applications such as file transfer, email and internet browsing are examples for non-real-time applications, for which performance requirements are usually limited to average message delay and throughput. These applications have very strict reliability requirements and even a small loss of the packets due to CRC errors or to congestion can have huge effects on the overall data-transfer performance. For example, a LAN that suffers only a 2% IP packet loss causes TCP [RFC793] throughput to drop to 47% of its value in the absence of losses [BPSK96]. In the case of a WAN that number drops to 23% effectively making TCP useless [Karn93]. Therefore, for applications using TCP it is much more critical to have reliable packet delivery, even though it might be delayed by a significant amount of time.

The characteristics of real-time communication applications differ significantly from those that are non-real-time [AKRS94]. As in real-time computing, the distinct feature of real-time communication is the dependency upon the times at which messages are successfully delivered to the recipient. Typically, the desired delivery time for each message across the network is bounded by a maximum delay, resulting in a deadline being associated with each message. This delay is imposed by the application's timing requirements. If a message arrives at the destination after its deadline has expired, its value to the end application may be greatly reduced. In some situations, messages that arrive late are discarded by the application. Streaming video is a perfect example of such an application [Mahanta97].

Some real-time applications do not care how much a prior to the deadline a message arrives. However, other applications consider early arrival harmful, and they require buffering at the receiver to achieve minimum jitter.

Applications that are interactive require a bound on jitter in addition to a bound on the delay. However, the jitter can be controlled at the receiver with buffering of the received data. The amount of the buffer space required can be determined from the peak rate and delay jitter of the delivery process and can be quite large for a network with no control of delay. For example [AKRS94], a video source transmitting 30 frames/sec, each frame containing 2Mb of data and experiencing a transmission jitter of 1sec, would require 60Mb of buffer space at the destination to eliminate the jitter. Another method of controlling jitter requires the timing of the packets in the network. It is a tradeoff between utilizing the buffers on the network, or at the destination of the stream.

In contrast to jitter guarantees, delay guarantees cannot be provided by buffering, because buffering can only delay the time of the delivery. Delay is a consequence of the statistical sharing that occurs in packet networks. Additional mechanisms are required for guaranteeing low delay in a real-time stream.

There are several mechanisms for supporting real-time communications. One of them is circuit-switching [AKRS94]. This type of transmission provides real-time delivery quite easily. However, for some types of real-time traffic, such as video [Mahanta97] or voice [KT01] [KBSSGM98] that are bursty in nature, this is accomplished at the expense of the bandwidth. Another type of mechanism for guaranteeing delay is giving preferential treatment for real-time packetized traffic over non-real-time traffic.

No real-time service can be provided if the client does not specify the requirements and the characteristics of the expected input traffic. These requirements are [AKRS94] [Ferrari90]:

- delay
- throughput
- reliability
- jitter

These characteristics represent performance bounds on the traffic which are easily quantifiable and in today's infrastructure are normally handled by the network. Other characteristics that cannot be easily expressed as bounds are:

- sequencing
- absence of duplications
- failure recovery
- service setup time

These characteristics of the traffic are normally handled by the transport protocols.

3 QoS

Most of the unpredictable and undifferentiated packet loss and jitter in today's networks result from the way in which the basic best-effort IP routers deal with the congestion [Armitage00]. A legacy IP router uses a FIFO queuing of packets destined for transmission on the associated outbound link. This queuing introduces latency (delay) and the potential for packet loss if the queue overflows. Then, the packets are discarded indiscriminately, regardless of the type of traffic they are carrying.

For applications that restrict their requirements of quality to the reliable delivery of the data, the transport-layer protocols, such as TCP [RFC793], make up for the inherent unreliability of packet transfer in IP [RFC791]; TCP reorders the sequence of packets, and detects and recovers from packet loss (or excessive delays) through a system of timers, acknowledgements, and sequence numbers. TCP also provides a credit-based flow control mechanism, which reacts to network congestion by reducing the rate at which packets are sent.

This type of queuing at the network layer and the recovery from the discarding of the packets at the transport layer is fine for elastic traffic, i.e., traffic such as email or file transfer that can adjust to large changes in delay and throughput (so long as the data eventually get there), but not for real-time traffic. The combination of IP/TCP does not have the mechanism required for satisfying the end-to-end timing requirements of an application. Other mechanisms are required for cooperating with the existing infrastructure of the Internet to provide guarantees of service according to the requirements of real-time traffic.

The term QoS refers to the performance of an IP packet flow through the network. The QoS parameters are defined by the application, and in the framework defined by the IETF they correspond to jitter, bandwidth, and latency. The end-to-end QoS is built upon the concatenation of QoS provided by each domain through which the traffic is going.

The artifacts of the best-effort routing can be reduced by partitioning the traffic into different groups that can be given different treatment appropriate to their performance needs. On high-speed backbone routers, where the processing and memory constraints of the hardware prevent the equipment from keeping track of the QoS requirements for each data flow, the trend is to partition the traffic based on the class of the stream that it is carrying, without identifying or prioritizing individual data flows. This type of partitioning the traffic is called DiffServ [DiffServ]. Another way of coping with the QoS requirements is to provide the means for each of the flows to reserve resources for each, or for a collection of a data flows. This type of mechanism is called IntServ [IntServ], and it is used in mostly in the access networks. Both of the methods for providing the QoS are presented in the sections below.

3.1 DiffServ

The main motivation behind DiffServ [RFC2475] is to decrease the complexity of the packet classification mechanism at the expense of the granularity with which it can isolate different classes of IP traffic. A common classification scheme is to pick N bits in a packet header to differentiate up to 2^N classes of packets. For each class a router has a classification rule that describes the router's processing of the packet belonging to a given class. The 2^N classes must equal or exceed the minimum number of differentiable traffic classes required by the underlying network.

The classification stage must keep up with the packet arrival rate similar to IntServ. If the classification stage is not keeping with the packet arrival rate, FIFO queuing occurs before the classification stage.

Router vendors differentiate their products based on cost versus performance. Hence usually, the number of classes that the router is capable of handling is proportional to the cost of the router. Moreover, the more classes the router is able to cope with, the more processing power the router requires.

In IPv4, the field ToS [RFC1349] is used to allow per-hop packet classification. The figure below shows the partition of the ToS field as it is described in the RFC.

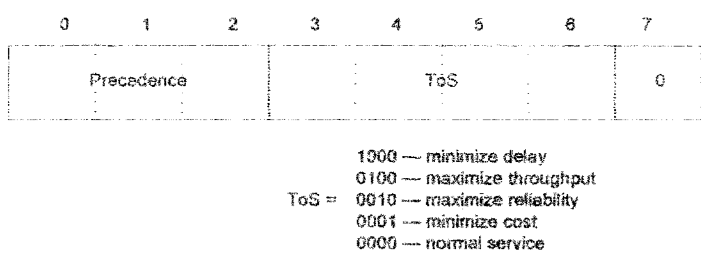

Figure 1: IPv4 ToS Field

Classification based on the Precedence field indicates relative temporal priority between 8 (2^3) precedence levels. RFC1812 [RFC1820] indicates that routes are to use increasing numerical values from 0 to 7 to indicate progressively higher forwarding priority. The figure above shows 5 of the 16 (2^4) ToS patterns and their usage. The remaining 11 possible values are currently not used.

The main drawback of the RFC1349 ToS field is that it imposes a very limited model for differentiated traffic handling. The Precedence field allows only relative priorities to be encoded. That is, packets of precedence 7 are transmitted before those of precedence 5, etc. In addition, the ToS subfield consumes half of the available bits for QoS based routing, regardless of the fact that this feature is rarely handled by the routers.

As the main motivation behind this architecture is to improve the speed of the classification of the packet at the cost of reducing the cardinality of the set of the possible classes, this architecture is best suitable for the backbone service provider space, where the router performance is at premium, and transport flow numbers count in hundred of thousands. The RFC1349 allows the repartitions of the ToS field as shown in the figure below, accommodating up to 2^6 different contexts.

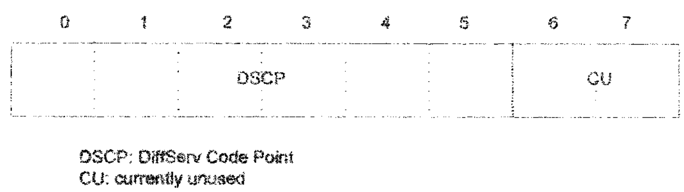

Figure 2: DiffServ Code Point

From the traffic management point of view, RFC2481 describes mechanisms for explicit congestion notification (ECN). This is accomplished by redefining the two CU from the DiffServ field as ECN Capable Transport (ECT) and Congestion Experienced (CE) bits. The ECT bit is set when both ends of the DiffServ flow support the processing of the CE bit. If the nodes do not support the explicit congestion notification, the CE bit is ignored. The ECN and CE bits are intended for use only on long-term congestion within the network.

DiffServ does not rely on resource reservation for satisfying the QoS requirements of the application, and it does not prioritize the packets based on the flow. The mapping between the flows and service classes is done either by statically pre-provisioning the network with SLA, or by making a DiffServ network RSVP aware.

In a network such as a Fixed Wireless Access Point to Multi-Point (FWA PMP) network, one sector covers a relatively low number of subscribers, whose aggregate bitrate is in the range of 2Mbps to 70Mbps depending on the distance from the service provider. In an environment with such a relatively low bitrate (by today's standards) it is still relatively easy to keep track of the QoS context of each of the data streams. Consequently, the IntServ architecture is much more suitable for such a system. Moreover, the scheduler presented in this thesis is not suitable for DiffServ deployed in a high-bandwidth environment, because it requires the modification of the header of the packet. In IP, such an operation requires a re-computation of the checksum, effectively defying the purpose of the DiffServ (which was created for providing QoS with absolutely minimum CPU resources).

3.2 IntServ

IntServ [RFC1633] was developed as an extension of the existing best effort routing currently implemented over the Internet. The primary motivation behind this work was to add a range of end-to-end services that would fulfill the requirements of real-time streaming and interactive applications. The compatibility with the existing Internet framework was also a requirement that the designers of this mechanism had in their mind.

IntServ implements an architecture requiring per-flow traffic handling at every hop along an application's end-to-end path and explicit signaling of each flow's requirements. RSVP [RFC2205] is the signaling protocol in the IntServ architecture [RFC2210].

Before a new flow is allowed to use a network's resources, it is subject to Admission Control by each router along the application's data flow path. A flow is admitted only when each router along the path indicates it can support the request. Admission is based on the source application's own characterization of its traffic profile, according to the application's QoS parameters. After a flow is admitted, the routers impose policing functions on the flow. These functions limit an application's ability to inject traffic in excess of its negotiated traffic profile. The figure below shows a reference model for the routers for an IntServ node.

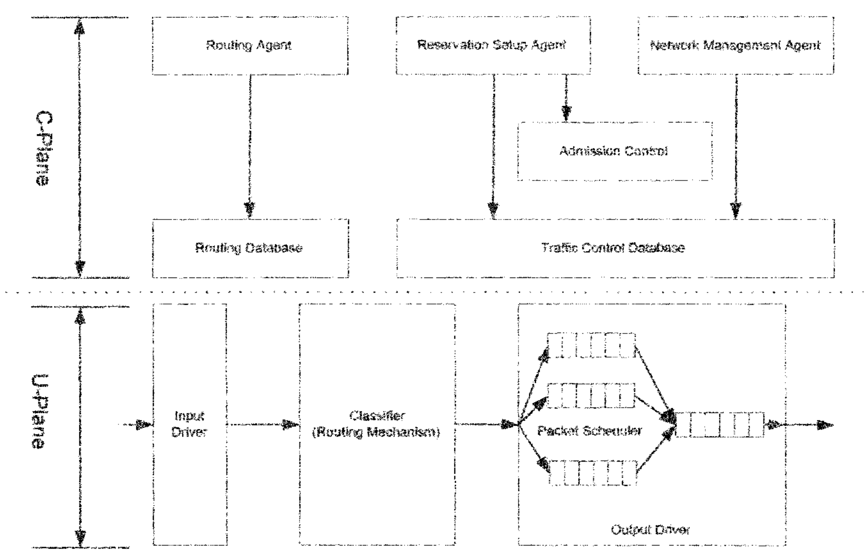

Figure 3: IntServ Architecture

- Routing Agent and the Routing Database are responsible for keeping track of the optimal routes to destinations on the network. The routing protocols include the widespread RIP, OSPF for smaller networks and BGP for core routers.
- Packet Scheduler manages the forwarding of different packet streams using a set of queues. The scheduling algorithm is usually based on Round-Robin, or WFQ.
- Classifier maps each packet onto a class. A class might correspond to a category of flows, or to a single flow, depending on whether the stream is unicast, broadcast or multicast. Normally, the mapping is established during the QoS negotiation phase.
- Admission Control implements the decision algorithm that a router or a host uses to determine whether a new flow can be granted the requested QoS without impacting earlier guarantees. Admission Control is invoked at each node to make a local accept/reject decision, at the time a host requests a QoS along some path through the network.
- Reservation Setup Agent implements the protocol used for setting up resource reservation according to the QoS parameters specified by the application. Currently, RSVP is the most widespread protocol for negotiating QoS requirements in the IntServ infrastructure of the Internet.
- Network Management Agent is responsible for administering the classifier and packet scheduler databases to set up controller link-sharing and to set admission control policies.

IntServ defines two service models [RFC1633], intended for real-time applications and elastic applications: controlled load (CL) [RFC2211] and guaranteed service (GS) [RFC2212]. The CL service is useful for allowing different application flows to share a network while constraining their degree of mutual interference. The CL service does not guarantee queuing delays or the bandwidth; it simply establishes a best effort tunnel through a network allowing different applications to share, while constraining their mutual interference. On the other hand, GS provides firm bounds on end-to-end datagram queuing delays. This makes it possible to provide a service that guarantees both delay and bandwidth. In this context, guaranteed service means that the datagrams arriving within a specified amount of time will not be discarded due to queue overflows, provided the flow's traffic stays within its specified QoS traffic parameters. This service is intended for applications that need a firm guarantee that a datagram will arrive no later than a certain time after it was transmitted from its source. During the QoS negotiation phase, an application provides a characterization of its expected traffic profile. The network calculates and returns an indication of the resulting end-to-end latency it can guarantee. If this latency is within the limits desired by the application, it can be assured that the path will deliver packets within this calculated latency bound. Otherwise, the application can modify its traffic characterization and request the network to recalculate the end-to-end latency that it is able to guarantee.

GS guarantees only the worst case latency. Due to the stringent traffic characterization, it is most suitable for intolerant real-time applications.

3.2.1 RSVP

The RSVP [RFC2205] [RFC2210] is the protocol used for the negotiation of the QoS parameters in the IntServ framework. It is used by a host to request a specific amount of bandwidth and prioritization of the traffic from the network in both unicast and multicast sessions. In the case of multicast sessions, it provides mechanisms to let the clients request different levels of QoS, effectively allowing the heterogeneity of the received data stream. The principal characteristics of RSVP are:

- RSVP is independent of the IP protocol version. It applies equally well to IPv4 and IPv6. Recent extensions have been made to RSVP for the establishment of MPLS paths [RFC3209].
- RSVP signaling messages follow the same routed path used by the U-plane traffic.
- RSVP is a receiver oriented protocol, i.e., the receiver of a data flow initiates and maintains the resource reservation used for that flow. The sources of the traffic use one pass with advertisement advertising its traffic's characteristics to the receivers. The receivers then tell the network which specific traffic signaling is required.
- RSVP signaling provides for a simplex reservation only. However, recent extensions to the protocol allow for duplex session establishment.
- RSVP uses soft state signaling. Routers along the traffic path must regularly be updated (refreshed) with information about active RSVP sessions. Otherwise, the router removes all states associated with the session, effectively tearing it down.

Recent extensions to the protocol [RFC2961] present mechanisms to reduce the number of messages required for refreshing the context of the session.

- RSVP messages are extensible and can carry additional objects relevant to the traffic and policy control entities at every router along the path between the source and the receiver.
- RSVP does not attempt to inform the routing protocol of new sessions appearing or of old sessions being removed. When a receiver requests the allocation of resources for a new flow, RSVP attempts to fulfill it with the resources that are still available along the flow's shortest path tree. The request will fail if no resources are available, possibly because some earlier reservations are still active, even if completely empty alternative paths exist in the network.

The main RSVP messages are PATH and RESV messages. Other message types include reservation confirmation message, error report message, and reservation and path tear down messages. In the RSVP specification, these messages are defined as follows: PATH, RESV, PATHErr, RESVErr, PATHTear, RESVTear, RESVConf.

An RSVP request consists of a flowspec and filter spec. Together, they form a flow descriptor. The flowspec specifies a desired QoS (bandwidth, delay, jitter, etc…). The filter spec defines the set of packets (the flow) to receive the QoS defined by the flowspec. The flowspec is used to set parameters in the node's packet scheduler, while the filter spec is used to set parameters in the packet classifier. Both packet classifier and packet scheduler are traffic control functions in the primary data packet forwarding path.

The packet classifier selects a service class for each packet in accordance with the reservation state setup by RSVP, while the packet scheduler implements QoS for each flow using one of the service models defined by the working group of the IETF [RFC2205]. The flowspec in a reservation request generally includes a service class (either GS or CL) and two sets of parameters: an Rspec (R for reserve) and Tspec (T for traffic). Both of these parameters describe the characteristics of the data stream.

RSVP includes two options that together are called "reservation style". One reservation option concerns the treatment of reservation for different senders within the same session: establish a distinct reservation for each upstream sender, or else make a single reservation that is shared among all packets of selected senders. The other option controls the selection of senders; it may be an explicit list of all selected senders, or a wildcard that implicitly selects all the senders to the session. In an explicit sender-selection reservation each filter spec must match exactly one sender, while in a wildcard sender-selection no filter spec is needed.

Sources that want to allow RSVP reservation for their session emit periodic PATH messages toward the session's destination. Each receiver host responds with RESV message upstream towards the sender. These messages create and maintain reservation state in each node along the path. RESV messages must be delivered to the sender, so that it can set up appropriate traffic control parameters for the first hop. At each intermediate node, two general actions are taken on the arrival of the RESV message:

1. Make a reservation; the request is passed to admission control and policy control. If either test fails, the reservation is rejected and RSVP returns an error message to the receiver. If both succeed, the node uses the flowspec to set up the packet

scheduler for the desired QoS and the filter spec to set the packet classifier to determine the QoS class for each data packet in the stream.

2. Forward the request upstream; the reservation request is propagated upstream toward the appropriate senders. The set of sender hosts to which a given reservation request is propagated is called the scope of that request.

It is important that the RESV messages are send to along the exact route taken by the PATH messages. The PATH message contains the following information:

1. Sender template: this object carries the format of the packets that the sender will originate. The template is the form of the filter spec that would be used to select the sender's packet from others in the same session on the same link.
2. Sender Tspec: this object defines the characteristics of the data flow that the sender will generate. Tspec is used by traffic control to prevent over-reservation, and unnecessary admission control failures.
3. Adspec: this object carries a package of One Pass With Advertisement (OPWA) advertising information. A received Adspec is passed to the local traffic control, which returns and updates Adspec. The updated version is then forwarded in PATH messages sent downstream. The end result is that the Adspec at the receiver end may be used to gather information for predicting the end-to-end QoS of the flow.

The figure below gives a simple example of an interaction between RSVP nodes during a QoS negotiation process for a simplex unicast connection.

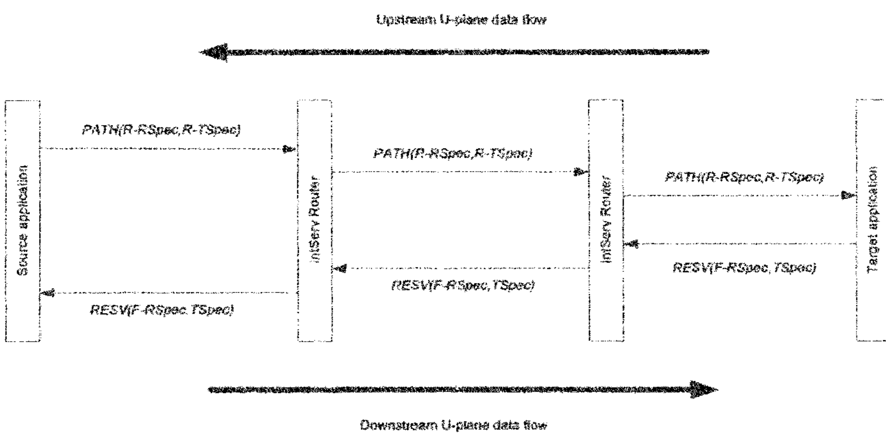

Figure 4: RSVP Signaling

The QoS negotiation process begins with the source application sending PATH message to the application that is going to receive the data. As the PATH message goes through the network, at each RSVP enabled node it is intercepted and modified to reflect the QoS capabilities of the route. The target application responds to the PATH messages with RESV messages identifying the session. Routers along the path correlate RESV messages with previously seen RESV messages, deciding whether resources can be successfully assigned to the session.

3.2.1.1 RSVP message formats

All the RSVP messages have a similar message header which specifies the message type and its length. The rest of the message contains RSVP objects.

The figure below shows a structure of the RSVP header. The purpose of the fields should be self-explanatory.

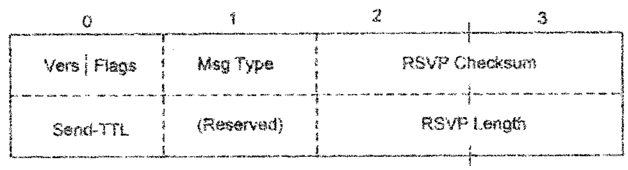

Figure 5: RSVP Message Format

3.2.1.2 Object formats

All RSVP objects have a common format which specifies the option and its length. The figure below shows a structure of an RSVP object.

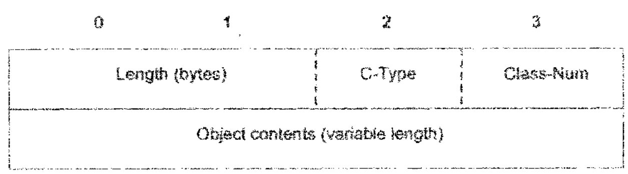

Figure 6: RSVP Object Format

Below is a short list of the most frequently used objects in RSVP messages:

Object	**Description**
Session	Session (destination of flow)
RSVP-Hop	Previous (PHOP) or next (NHOP) hop address
Time-values	Refresh period of the path or reservation messages
Style	Reservation style and the style-specific information
Source-Tspec	Upper bound on traffic that source application generates
Source-Rspec	The desired QoS
filter spec	Filter which defines packets receiving QoS
flowspec (RSpec, TSpec)	The flow specification: QoS and traffic parameters
Source-template	Source application identifier (transport-level address)
Adspec	A summary of advertised QoS capabilities on path
Error-spec	Error (or confirmation) specification

Policy-data	Opaque data that is handed to policy control module
Integrity	Keyed MD5 authentication and verification data
Scope	A list of upstream routers used to prevent loops
RESV-Confirm	Address where to send the reservation confirmation

Table 1: Most Frequently Used RSVP Objects

3.3 Packet Scheduling

Packet scheduling disciplines, together with admission control algorithms, provide the two most important components in the mechanism for providing guaranteed service to the clients [Zhang95]. While admission control reserves resources during connection establishment, packet scheduling disciplines allocate resources according to the reservation during data transfer. Three types of resources are allocated by service disciplines:

- bandwidth (which packets get transmitted)
- delay (when do those packets get transmitted)
- buffer space (which packets get discarded)

The management of the buffer space and the delay adds a certain level of elasticity to the traffic in the case of network congestion, allowing for priority queuing. However, the queuing adds delays, effectively degrading the timing requirements of the application.

Non real-time traffic tolerates latency better than packet loss (for example TCP traffic carrying email). In such cases the packet scheduler should be allowed to buffer the packet for a longer period of time, implying longer queues. However, other types of traffic, such as real-time audio, prefer that packets be discarded if held too long by the network. For this type of behavior shorter queues are better.

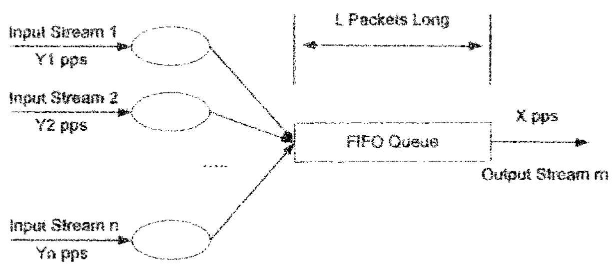

Figure 7: Simple Packet Scheduler

For example [Armitage00], let packets arrive from each input port in the figure above at a maximum rate of Y_1 through Y_n packets per second (pps). The outbound link extracts packets from the queues at X pps. The total input rate Y is the sum of each individual stream ($Y_1 + Y_2 + \ldots Y_n$). When Y < X, packets will not need to wait in the queues. However, it is more than likely that Y, during periods of congestion, can burst well above X. When this happens, the queue will grow in size. The number of packets P in the queue after some interval T is expressed as $P = T\,(Y-X)$. If a packet arrives at time T when the queue is partially full, it will experience an additional latency of X times P seconds (because the packet must wait for the queue to drain at X pps). If a packet arrives when the queue is full, that is when $P = L$, the packet is going to be dropped.

Since the queues in the routers exist to absorb the traffic bursts without loss of the data, packet scheduler should ensure that queue's average occupancy is kept low. If the queue operates with a high average occupancy, the space available for the buffering of additional packets is reduced. This greatly reduces the elasticity of the queue, and increases the buffer requirements of the scheduler. Most importantly for the real-time type of traffic, the average latency experienced by the traffic sharing a given queue increases as the average queue occupancy increases. So keeping the occupancy down improves the end-to-end latency.

If the traffic is partitioned into different flows or classes of traffic (depending on whether we are in the context of the DiffServ or IntServ framework), each flow/class is mapped onto a queue that satisfies the latency, jitter, and packet loss QoS characteristics. The queues are emptied in an order that depends on the scheduling discipline implemented by the router.

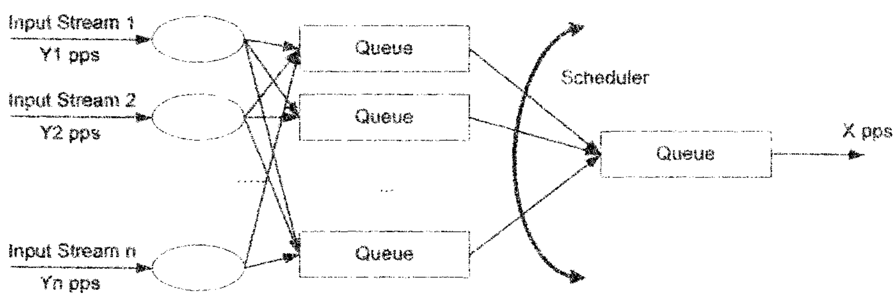

Figure 8: Example of a Queuing System

The scheduling policy can be classified as either work-conserving or non-work-conserving [AKRS94]. A scheduler is work-conserving if an output link will never be idle as long as there are packets waiting to use that link. Work conservation promises lower average end-to-end delays for packets. The problem lies in the fact that due to the surges in the load on the network, with the work conserving disciplines the traffic usually acquires an additional jitter.

Methods which minimize jitter require buffering of the packet even when the server is idle. Unfortunately, this increases the average delay of all packets and decreases the throughput of the server [Zhang95].

3.3.1 Strict Priority

A strict priority scheduler orders the queues by descending priority and servicing a queue at a given priority level only if all queues of higher priority levels are empty. The side effect of this behavior is the starvation of the lower priority queues. The starvation can be avoided by a careful provisioning of the system, making sure that different traffic classes are never allowed to exceed a given fraction of the link capacity. This will allow the scheduler to empty higher priority queues every so often, giving the scheduler time to service lower priority queues.

This scheduler has a worst case latency that depends only on the speed of the link.

3.3.2 Round-Robin

Round-Robin avoids queue starvation by cycling through the queues one after the other. It transmits one packet before moving on to the next queue. If some queues are empty, the other queues are serviced more frequently. The drawback of RR is that latency bounds are hard to enforce; the service interval depends on how many other queues have packets in them at any instant, and their length. If we have N queues (one for each priority), a link with speed L, it will take P/L time to transmit a packet. So, if the scheduler skips a queue, the packet will have to wait (N-1)*(P/L) until the scheduler comes back. This behavior generates jitter.

3.3.3 Deficit Round-Robin

Deficit Round-Robin [SV95] is an extension to the Round-Robin algorithm. It assigns each queue a constant Q_N reflecting the long-term average number of bytes per scheduling cycle that we want the queue to send, and a variable D_N, which is a difference between Q_N and B, the actual number of bytes sent. The bytes are transmitted from the queue N if there are packets waiting to be sent, and if $Q_N + D_N >= B +$ (the bytes in the next packet on the queue). If the queue is empty, D_N is set to zero. However, if the scheduler stops before emptying the queue, a deficit exists between the number of bytes scheduled for sending, and the number of bytes that were actually sent. In this case, the deficit is reset to $(Q_N + D_N - B)$. For each queue, $Q_N + D_N$ is the maximum number of bytes that can be sent during a service interval of the queue. Recalculating D_N at the end of a service interval compensates the queue next time with more bytes to send if it was short-changed this time.

Assuming packets are never larger than the MTU of the link, D_N is never larger than the MTU. After the scheduler has cycled through all N queues K times, the expected number of bytes sent by the queue N is K x Q_N. This value never exceeds the actual number of bytes sent by more than MTU. Each queue may be assigned different values for Q_N, leading to different long-term relative bandwidth allocations. If less than N queues have packets to send, the link's bandwidth is dividing according to the Q_N parameters of the queues with packets.

3.3.4 Weighted Round-Robin

This algorithm uses the same concept of quantum and deficit, but the queue is serviced until a packet is sent that put the number of bytes sent (B) over the allowed limit for the queue (which is still $Q_N + D_N$). However, the deficit is a value that reflects the amount sent in excess of $Q_N + D_N$ and acts to decrease the number of bytes the queue may transmit the next time around. As with the deficit round-robin, the quantum for each queue can be used to establish proportional bandwidth sharing.

3.3.5 Fair Queuing/Virtual Clock

Fair queuing/virtual clock [Zhang95] attempts to approximate the event link sharing that would result if packets from each queue were being evenly interleaved on a bit-by-bit basis just as in a TDM system. The scheduler polls the queues in a round-robin fashion pulling one bit from each queue during a single cycle. This behavior ensures that whenever N queues have data to send, each queue is serviced at 1/N of the link's rate. A packet is fully transmitted when its last bit is removed from the queue. Over multiple packets, this algorithm empties each queue at approximately the same rate as it would have been transmitted by using a TDM link. Because the scheduler keeps track of the number of bits sent, sharing between queues is fair.

3.3.6 Weighed Fair Queuing

WFQ is a variation of round-robin that continuously recalculates the priority so that the next queue to be serviced is the one that needs to be serviced to meet its long term average bandwidth target. This algorithm is also known as PGPS (packet generalized processor sharing) [Zhang95]. This algorithm allows different weights to be applied to individual queues. Whenever N queues with data to send are present, each queue M is serviced to give it a fraction Y_M of the link's rate (where $Y_1 + Y_2 + \ldots Y_n =$

1). When some of the queues are empty, excess link capacity can generally be distributed among the remaining queues according to their relative weights.

3.3.7 Delay Earliest Due Date

In this algorithm, the scheduler keeps a deadline for each packet [Zhang95]. The deadline of the packet is the sum of its arrival time and the period of the traffic stream. The packets are then queued and sent in the order of the increasing deadlines. However, since this scheduler takes into account the period of the traffic stream, it does not work that well for bursty traffic.

3.3.8 FIFO+

FIFO+ [CSZ92] attempts to reduce the jitter by keeping track of the average delay seen by packets in each flow. This is done by computing the difference between the packet's particular delay and the average delay for the stream. This difference is added (or subtracted) to the value accumulated by the packet during its trip through the successive nodes in its path. The scheduler then prioritizes the packet based on the offset between the field in the header, and the average for the stream.

3.3.9 Jitter Earliest Due Date

The jitter earliest due date [VZF91] service discipline is an example of a non-work conserving scheduler. This discipline extends the delay earliest due date scheduler to provide bounds on the maximum delay different between two packets. After a packet has been serviced at each server, a field in its header is stamped with the difference between its deadline and the actual finish time. Unlike in FIFO+, a delay buffer at the next server holds the packet for this period before it attempts to schedule it.

3.3.10 Lead-Time Scheduling

The lead-time scheduling discipline [DLS00] keeps track of the customer lead-times, where the lead-time is defined as the time remaining until the deadline elapses. Let $leadTime_{i,j}$ be the lead-time of the i^{th} packet in the stream j, then the lead-time is defined as

$$leadTime_{i,j} = deadline_{i,j} - currentTime\,.$$

The lead-time determines the relative priority of the packet in the queue q as follows

$$p_q = \min\{leadTime_{i,j,q}\}$$

An interesting feature of this scheduler is the fact that the customer lead-times decrease linearly while a customer is in the queue, leading to the reevaluation of priorities during each round of the scheduler.

3.3.11 Other Non-Work Conserving Schedulers

Zhang [Zhang95] in his survey presents other types of non-work-conserving schedulers; stop-and-go, hierarchical round-robin, and rate-controller static priority. The goal of these schedulers is to provide an end-to-end bound on the jitter of the traffic. In all these algorithms, this is accomplished by buffering the packet at each of the nodes along the path by an amount of time that reduces the variance of the traffic. However, introducing the delay at each of the nodes introduces additional memory requirements at each of the servers, and it also reduces the efficiency of the server.

3.4 RTP

At the transport layer, TCP might be appropriate for non-real-time applications. However, for real-time traffic where time constraints are quite stringent, and where the packet loss causes TCP to scale down the sliding window and retransmit the lost packets, another transport layer protocol is required for fulfilling the timing requirements of the applications.

RTP [RFC1889] provides end-to-end delivery services for data with real-time characteristics. The services provided by this protocol include pay-load type identification, sequence numbering, time stamping and delivery monitoring. RTP is used on top of UDP to make use of its port number identification and checksum.

RTP supports data transfer to multiple destinations using multicast distribution if provided by the underlying network. RTP itself does not provide any mechanism to ensure timely delivery or provide other QoS guarantees, but relies on lower level services to do so. It does not guarantee delivery or prevent out-of-order delivery. It also does not assume that the underlying network is reliable and delivers packets in sequence. RTP, similarly to other transport protocols, relies on RSVP to perform resource reservation and establish the path between the source and the sink of the stream.

RTP consists of two parts: the data part to carry the user (U-plane) data, and a control (C-plane) part for monitoring the QoS characteristics of the connection and to convey information about the participants in the session.

3.4.1 RTP Data

RTP data packets consist of a 12 byte header followed by the payload. The main fields of the header are given below:

- Payload type; it is a one byte field that identifies the type of the payload in the packet, for example an audio stream. Having payload type in every packet avoids connection setup and permits dynamic changes of encoding/decoding of the stream.
- Timestamp; it is a 4 byte field describing the time instance of the generation of the packet. The timestamp depends on the payload time. For example, a number of speech or video packets might have the same timestamp if they belong to the same burst.
- Sequence number; it is a 20 bit number that allows the sequencing of the stream within the same timestamp, and detection of loss of the packets.
- Marker bit; the meaning of this bit depends on the payload type. For video, it marks the end of a frame and for audio the beginning of a speech burst.
- Synchronization source identifier (SSRC); this field identifies the synchronization source. This identifier is chosen at random with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier.

* Contributing Source Identifier (CSRC); the CSRC lists the contributing sources for the payload contained in this packet. CSRC fields are inserted by the mixers.

3.4.2 RTP Control

RTP Control Protocol (RTCP) [RFC1889] is based on the periodic transmission of control packets to all participants in the session, using the same distribution as the data packets. RTCP performs the following four functions:

1. Provides feedback on the quality of the data. This information is used in the flow and congestion control functions.
2. It carries a persistent transport level identifier for an RTP source called the canonical name (CNAME). The receivers require the CNAME to keep track of each participant if a conflict is discovered, or a program is restarted. Receivers also require the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video.
3. It keeps track of the number of participants in the session so that it can control the rate as RTP scales to a larger number of participants.
4. The last, optional function is to convey session control information, for example participant identification to be displayed in the user interface.

3.5 Discussion

There are a few mechanisms that exist within today's framework of the Internet which together cooperate to provide an end-to-end QoS. Many distinct architectures have been proposed in academia and implemented to a limited extent in the laboratory environment [ACH98]. The most widely accepted IntServ and DiffServ have extended the existing infrastructure of the Internet with the partitioning of the traffic into distinct classes of service that can be prioritized based on the QoS specifications of the application. Whereas DiffServ does not use signaling to map traffic flows onto separate classes of service, IntServ relies on explicitly reserving resources at each node along the path of the flow.

The partitioning of the traffic into different classes of service introduced by IntServ and DiffServ has encouraged the development of a wide range of scheduling algorithms. They are classified into work-conserving and non-work-conserving schedulers. While the first emphasize fair allocation of the resources of the server to different flows of traffic, the latter concentrate on minimizing the distortion to the properties of the traffic as it traverses the network by regulating the delay at each server. It has been observed that smaller end-to-end delay bounds can be obtained by talking into account the delay dependencies among successive switches traversed by the connection. FIFO+ and the lead-time schedulers are examples of such schedulers, although in contrast to the lead-time scheduler, the primary objective of FIFO+ is to reduce the jitter.

At the level of the transport layer protocols, RTP is the most widespread protocol for specifying the temporal relevance of the data. It cooperates with RSVP for establishing and maintaining QoS required by the application. However, since RTP is a transport layer protocol, it cannot directly cooperate with the network to enhance the QoS at the intermediate routers along the path of the connection.

4 Deadline Distribution for Lead-Time Scheduling

As it was mentioned in the introduction, the analysis of the lead-time scheduling discipline presented by Doytchinov et al [DLS00] is the most detailed one available today. However, their analysis is limited to a uniform deadline distribution for the EDF queuing discipline. On the wireless network simulated in this experiment, the majority of the customers are clustered at the very last node of the system, causing the deadline distribution of the traffic to have a more exponential-like shape. This section presents a derivation of the theoretical frontier distribution for a uniform deadline distribution. The results obtained in this section complete to the results obtained by Doytchinov et al.

Definition 1: The frontier $F(t)$ is the largest lead-time of any customer who has ever been in service, whether still present or not [DLS00].

Frontier for the nth queuing system is defined as[1]:

$$F_n = H_n^{-1}\left(\frac{Q^n}{\lambda}\right)$$

where Q^n is the queue length of the queuing system n, λ is the customer inter-arrival time, and H_n is defined as[2]:

$$H_n(x) = \sqrt{n}H\left(\frac{x}{\sqrt{n}}\right) = \int_x^{\infty}(1 - G_n(t))dt$$

[1] [DLS00], equation 4.6
[2] [DLS00], equation 4.2 for x > 0

where the cumulative distribution function of the lead-times in the n[th] queuing system G_n is[3]:

$$G_n(x) = P\{L_n^{(n)} \le x\}$$

$G_n(x)$ is the cdf based on a uniform density $U(A,B)$ uniform on $[A,B]$[4] shown in the figure below.

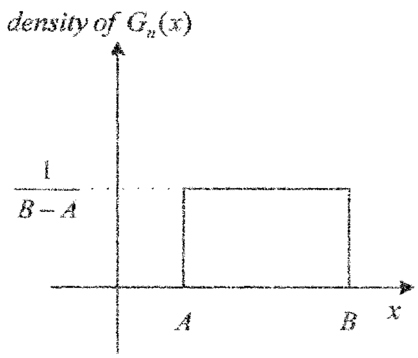

Figure 9: Plot of the Density U(A,B) of G_n

$G_n(x)$ is an integration of the density function. The figure below shows $G_n(x)$.

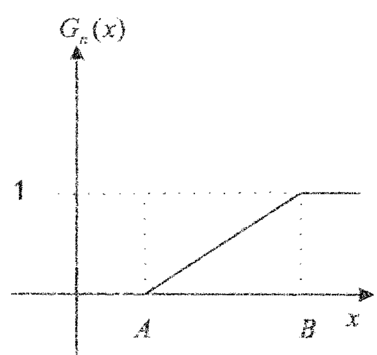

Figure 10: Plot of the cdf G_n

[3] [DLS00], equation 4.1
[4] [DLS00], section 4.1

In order to obtain H_n we have to find $1-G_n(t)$. The resulting distribution is shown in the figure below.

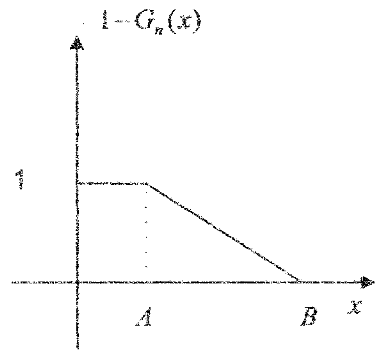

Figure 11: Plot of 1 - G_n

The equation for $1-G_n(x)$ can be written as

$$1-G_n(x)=I_{[0,A]}+\left(1-\frac{x-A}{B-A}\right)I_{[A,B]}$$

In order to find the distribution of the frontier, we have to find the inverse of H_n, where

H_n is given by

$$H_n(x)=\int_x^{\infty}(1-G_n(t))dt$$

H_n depends on G_n , which has three distinct cases as shown in the figure below:

Figure 12: Regions of H_n

Case 1: $x \le A$

$$H_n(x) = \int_x^A dt + \int_A^B \left(1 - \frac{t-A}{B-A} I_{[A,B]}(t)\right) dt$$

$$= A - x + (B-A) - \frac{1}{B-A}\left(\frac{t^2}{2} - At\right)\Big|_A^B = B - x - \frac{1}{B-A}\left(\frac{B^2}{2} - AB - \left(\frac{A^2}{2} - A^2\right)\right)$$

$$= B - x - \frac{1}{B-A}\left(\frac{(B-A)(B+A)}{2} - A(B-A)\right)$$

$$= B - x - \frac{1}{2}(A+B) + A$$

Case 2: $A < x < B$

$$H_n(x) = \int_x^B \frac{t-A}{B-A} I_{[A,B]}(t) dt$$

$$= B - x - \frac{1}{B-A}\left(\frac{t^2}{2} - At\right)\Big|_x^B = B - x - \frac{1}{B-A}\left(\frac{B^2 - x^2}{2} - A(B-x)\right)$$

$$= (B-x)\left(1 + \frac{A}{B-A}\right) - \frac{B^2 - x^2}{2(B-A)} = (B-x)\left(\frac{B}{B-A}\right) - \frac{B^2 - x^2}{2(B-A)}$$

$$= \frac{1}{2(B-A)}(2B(B-x) - (B^2 - x^2) = \frac{1}{2(B-A)}(2B^2 - 2Bx - B^2 + x^2)$$

$$= \frac{1}{2(B-A)}(B^2 - 2Bx + x^2)$$

$$= \frac{(B-x)^2}{2(B-A)}$$

Case 3: $x \geq B$

$$\text{For } x > B \qquad H_n(x) = 0$$

Therefore, putting all the three cases together we get

$$H_n(x) = \begin{cases} \dfrac{A+B}{2} - x & x < a \\[2ex] \dfrac{(x-B)^2}{2(B-A)} & A \leq x \leq B \\[2ex] 0 & B < x \end{cases}$$

The figure below shows the graph of H_n

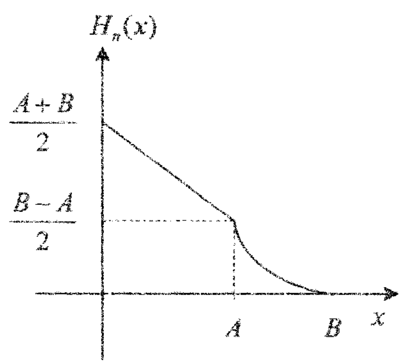

Figure 13: Plot of H_n

Now we find $H_n^{-1}(y)$. Again, we have three distinct cases.

Case 1: $\dfrac{B-A}{2} < y \leq \dfrac{A+B}{2}$

$$\frac{A+B}{2} - x = y$$

Isolating x we get

$$x = \frac{A+B}{2} - y.$$

Case 2: $0 \le y < \frac{B-A}{2}$

$$\frac{(x-B)^2}{2(B-A)} = y.$$

Isolating x we get

$$x - B = \pm\sqrt{2(B-A)y}$$

$$x = B - \sqrt{2(B-A)y}.$$

Case 3: $y \ge \frac{A+B}{2}$

From the graph, we can see that for this region $x = 0$. Thus putting all the three cases together we get

$$F_n(y) = H_n^{-1}(y) = \begin{cases} 0 & y \ge \frac{A+B}{2} \\ \frac{A+B}{2} - y & \frac{A+B}{2} > y \ge \frac{B-A}{2} \\ B - \sqrt{2(B-A)y} & \frac{B-A}{2} > y \ge 0 \end{cases}$$

Plotting F_n we get the following figure

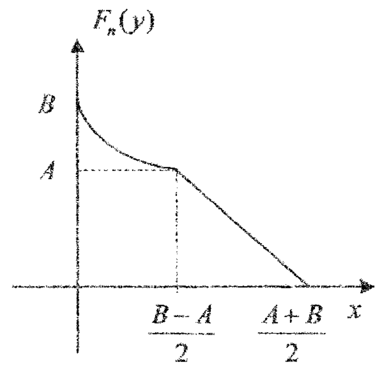

Figure 14: Plot of F_n

Now, let $y \therefore w = \frac{Q}{\lambda}$, and let F_{th} denote the theoretical frontier distribution. From the previous equations we get

$$F_{th}(x) = 1 - \frac{\lambda}{Q_n} H_n(x \vee F_n) \qquad \text{for } x > 0$$

$$= 1 - \frac{1}{w} H_n(x \vee F_n)$$

Case 1: $w \geq \frac{B-A}{2} \Rightarrow F_n \leq A$

This case falls into the region to the left in the figure below.

Figure 15: Regions of F_n

Case 1a: $x \le F \Rightarrow x \vee F = \max(x, F) = F$

$$H_n(x \vee F_n) = H_n(F_n(w)) = w$$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(x \vee F_n) = 1 - \frac{1}{w} w = 0$$

Case 1b: $F < x < A \Rightarrow x \vee F = x$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(x) = 1 - \frac{1}{w}\left(\frac{A+B}{2} - x\right)$$

Case 1c: $A \le x \le B \Rightarrow x \vee F = x$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(x) = 1 - \frac{1}{w}\left(\frac{(x-B)^2}{2(B-A)}\right)$$

Putting all the three cases together we get:

$$F_{th}(x) = \begin{cases} 0 & x \le F \\ 1 - \dfrac{1}{w}\left(\dfrac{A+B}{2} - x\right) & F < x < A \\ 1 - \dfrac{1}{w}\left(\dfrac{(x-B)^2}{2(B-A)}\right) & A \le x \le B \end{cases}$$

Case 2: $w < \dfrac{B-A}{2}$

Case 2a: $x \le F \Rightarrow x \vee F = F$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(F_n(w)) = 0$$

Case 2b: $F < x < B \Rightarrow x \vee F = x$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(x) = 1 - \frac{1}{w} \frac{(x-B)^2}{2(B-A)}$$

Case 2c: $B \leq x \Rightarrow x \vee F = x$

$$F_{th}(x) = 1 - \frac{1}{w} H_n(x) = 1$$

Putting them together we get the following:

$$F_{th}(x) = \begin{cases} 0 & x \leq F \\ 1 - \frac{1}{w} \frac{(x-B)^2}{2(B-A)} & F < x < B \\ 1 & B \leq x \end{cases}$$

Both results for F_{th} correspond to the results presented by Doytchinov et al [DLS00][5].

[5] [DLS00], section 4.1

5 Problem Statement

As it was presented in the survey of the scheduling algorithms, the priorities in the lead-time scheduling algorithm are governed by the following equation

$$leadTime_{i,j} = deadline_{i,j} - currentTime$$

where i and j denote the packet number and the stream number to which the packet belongs. In order for the lead-time scheduling to work in an IntServ multi-hop environment, this equation has to be extended to the following form

$$leadTime_{i,j,k} = deadline_{i,j,k} - currentTime_k - remainingTripTime_k$$

where the extra subscript k identifies the node router in the number of N routers which the flow has to travel, and the extra parameter $remainingTripTime_k$ is the minimum time required for the packet to go through the $N - k$ remaining routers. The result is the maximum amount of time that the packet can spend waiting to be sent in the router k. This information has to be available at each node through which the flow is going.

RSVP provides for admission control of the traffic with given QoS requirements up to a point when the network cannot accommodate the new flows. However, there are no mechanisms in the traffic flow itself to dynamically adjust the priorities of the flows based on their timing requirements that might be influenced by the ever changing load on the network. Transport layer protocols, such as RTP, do have timing information about the temporal relevance of the stream. However, this information is not available to the routers at the network layer protocols. The most obvious reason for this is that the overwhelming majority of the internet infrastructure is based on IPv4, which inherently does not have support for carrying the time information about the packet.

This thesis studies the feasibility of the implementation of the lead-time scheduling algorithm in a typical IntServ network. The thesis looks specifically into the possible issues that arise at the level of signaling (RSVP), and scheduling.

The secondary objective of this thesis is to simulate the lead-time scheduling algorithm. The simulation is done in the context of an FWA PMP network. The simulation does not take into account the C-plane signaling. The simulated network is configured statically for the whole duration of the simulation. The results are compared with the results obtained using FIFO, and strict-priority scheduling in the same network configurations.

6 Support for Lead-Time Scheduling in IntServ

In order for the lead-time scheduling algorithm to work in the IntServ framework, we need to have mechanisms in the packet scheduler and the signaling protocol to take into account the timing information of the flow.

6.1 Support for Lead-Time Scheduling in IP

IPv4 does have an 8-bit field called TTL. The purpose of this field is to make sure that the packets are not routed indefinitely on the network in case there is a loop in the route. As the packet goes through the network, each time it passes through a router this field gets decremented. When the value of this field drops to zero, the packet is discarded. An 8-bit field allows the packet to have up to 2^8 hops in its route.

One possibility for storing the lead-time, deadline, or a timestamp of the packet could be to use the TTL field. As it was explained in the case of the voice traffic, the trip time for voice packets should be no more than 100-150ms. Therefore, to use this field for the worst case scenario, it would be required to introduce a scaling factor for the time value. In order to cover the worst acceptable trip time, this factor would have to be 3, implying that this would be the best resolution of the timer. This might lead to large cumulative errors when the same part of the network is routing traffic for short connections (connections with only a few hops in between) and long connections.

For smaller networks where both ends of the connections are not far away (in terms of the number of hops), for example in the case of large corporate networks or closed networks, the solution of using the TTL field in the IP header might still be an acceptable solution. However, this means that there would have to be two disjoint solutions for small and large networks, causing inconsistencies in the same protocol for networks of different sizes.

Another possibility for computing the lead-time of the packet is to use the Internet Timestamp option of the IP header [RFC791].

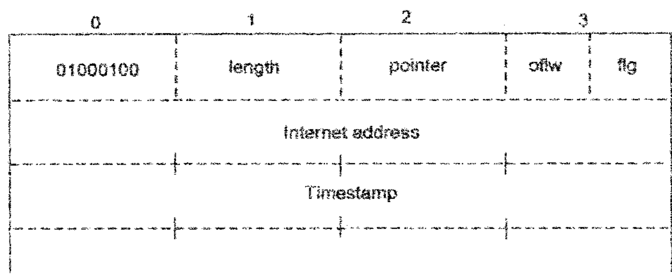

Figure 16: IPv4 Header

This option records the time (in milliseconds) at each host that the packet is going through. This time is usually counted since time zero for UNIX, commonly refered as UT. If the routers cannot provide the time in milliseconds, or cannot provide the time with respect to UT, then any time may be inserted as a timestamp provided the high order bit of the timestamp field is set to one indicate the use of non-standard value.

Just as in the case of the TTL field, the usefulness of this option is limited by its size; the field "length" holds the length of the whole option field (in bytes). Although the length of this field is 8-bits, the specification of IP [RFC791] limits the length to a maximum of 40 bytes. This gives enough space for keeping track of only five timestamps. However, for the purposes of computing the lead-time, we only need to know the time at which the packet has left the source, so only one timestamp would be required.

6.2 Support for Lead-Time Scheduling in RSVP

Another parameter needed for computing the lead-time is the trip time of the remaining route. As it was explained in the earlier sections, RSVP is perfectly suited for collecting and disseminating the remaining trip timer parameter required by the schedulers. In the basic RSVP reservation model [RFC2205]; a receiver sends a reservation request upstream, and each node in the path either accepts or rejects the request.

When a receiver generates an RESV message to be sent for a GS reservation request, it must include two additional objects, LTSTIMESTAMP and LTSTRIPTIME. Following the format of the RSVP object formats presented in section 3.2.1.2, LTSTIMESTAMP contains the 32 middle bits of the NTP time, identifying the instance in time when the RSVP packet has been sent. The object LTSTRIPTIME contains a 32 bit value denoting the cumulative trip time of the RESV message. At each RSVP enabled router k, the $\mathrm{LTSTRIPTIME}_k$ is computed from the difference of the local NTP provided time and the $\mathrm{LTSTIMESTAMP}_{k-1}$. This value corresponds to the $remainingTripTime_k$ parameter introduced in section 5, which is taken into account by the local lead-time scheduler when computing the $leadTime_k$ for each packet for each flow. $\mathrm{LTSTRIPTIME}_k$ is computed during the session establishment and made subsequently available to the scheduler for computing $remainingTripTime_k$ for the U-plane traffic.

This way of computing $remainingTripTime_k$ gives meaningful estimates of the remaining trip time only if the RESV messages are processed at the same priority as the

U-plane traffic. Otherwise, if the RESV is processed at lower priority, the estimate will be larger than the $remainingTripTime_k$ for the U-plane traffic, and vice-versa; if the RESV is processed at higher priority than the U-plane traffic, the estimate will be smaller.

Since RSVP is based on the concept of soft state, $remainingTripTime_k$ is be recomputed every state refresh cycle, effectively adapting to the changing conditions of the network.

The accuracy of $\mathrm{LTSTRIPTIME}_k$ and $\mathrm{LTSTIMESTAMP}_k$ and ultimately $remainingTripTime_k$ depends on the accuracy with which the lead-time scheduling enabled nodes synchronize their network time with protocols such as NTP. In order for the real-time voice traffic to have acceptable QoS characteristics, the maximum packet RTT *mxRTT* has to be less than 250ms [KBSSGM98] [KT01]. The cumulative error introduced by the drifting router times at the destination node *d* has to be less than

$$\xi = mxRTT - remainingTripTime_d \,.$$

If the cumulative error is greater than ξ, the user will perceive an echo.

Taking figure 4 presented in the section 3.2.1 as an example, a lead-time scheduling router (let's call it router *k*) has to perform the following sequence of operations for computing the parameters for the lead-time scheduler:

- Extract the object LTSTIMESTAMP from the received RESV message
- Compute $\mathrm{LTSTRIPTIME}_k$ according to

  $\mathrm{LTSTRIPTIME}_k = \mathrm{LTSTRIPTIME}_{k+1} + currentTime_k - \mathrm{LTSTIMESTAMP}_{k+1}$
- Set $remainingTripTime_k$ of the scheduler to $\mathrm{LTSTRIPTIME}_k$.

- Set $LTSTIMESTAMP_k$ to *currentTime*$_k$
- Make $LTSTRIPTIME_k$ and $LTSTIMESTAMP_k$ available to the Reservation Setup Agent for the construction of the RESV message for the remaining reservation trip.

Each router has to repeat the steps outlined above.

7 Simulation Environment

Most of the existing research in the area of the wireless solutions has been concentrated in wireless LANs, cellular networks, or satellite networks. The former of these systems are extremely short range solutions (less than 5Km) and the satellite networks are the other extreme of the spectrum (in the range of 36000Km for the geo-stationary orbit). The terrestrial wireless access networks typically have the range from 20Km to 700Km (in the latter case, the solution might require repeaters, introducing additional delays). The difference in the range, along with the considerable difference in the bandwidth, packet-delay, and consequently QoS characteristics present enough interest to justify studying a fixed wireless network in more detail.

The purpose of this simulation is to investigate the impact of different scheduling algorithms, including lead-time scheduling, on the delay and jitter of the voice traffic.

Although there are considerable differences at the MAC layer between the two most popular technologies, CDMA and TDMA [Karn93] [KAZ98], this thesis does not take into account these details. The advantages of one over the other usually show up in a very noisy environment, and not in the delay characteristics of the packets. Therefore, this topic is outside the scope of this thesis.

The fixed wireless access networks are used in the following applications:

- To serve new growth along existing routes where copper facilities have been exhausted. Reinforcement costs along existing routes are often very expensive and radio facilities may be appropriate based upon economic evaluations.

- To serve areas where no facilities currently exist to respond to customer's order. For example, radio would be used to avoid extending existing facilities that currently do not reach new housing developments.
- To serve areas where existing service, provided on a wire or radio facility reaches the end of its service life or it provides a non acceptable service grade. Plant replacements are expected in areas where open wire currently provides a non-acceptable quality of service, due to age, lightning problems, AC induction, etc.
- To provide emergency restoration facilities, temporary service for one-time events, construction contractors or for the purpose of postponing long-term cable projects until it is certain that development will actually occur. This type of application is particularly useful when a radio system has already been installed in an area to provide services in applications mentioned above.
- To bring telephone services to remote area that cannot be reached economically with wires/fiber.

Moreover, with the invention of the Internet, the PMP FWA, just like the traffic carriers, have evolved from providing POTS into packet networks converging both the data and real-time voice traffic onto the radio channels. This presents quite a challenge for these networks from the QoS perspective.

A very popular access network technology available to service providers competing with a PMP FWA network is a cellular network. In such a network the end user gains mobility, at the expense of bandwidth. This drawback directly impacts the QoS, and limits the possible usage of the technology to low-bandwidth applications, such as compressed voice. Recent developments in the 3G standards have greatly expanded the available bandwidth. However, the high costs related to the upgrade of the current infrastructure make this technology out of reach for the near future. Another drawback of the cellular technology is a very short coverage area of the base station. The usual range of one base station in a cell network is less than 5Km, whereas in a fixed wireless access network the range extends up to 700Km.

The figure below shows a modern fixed wireless PMP network in the context of today's telecom infrastructure.

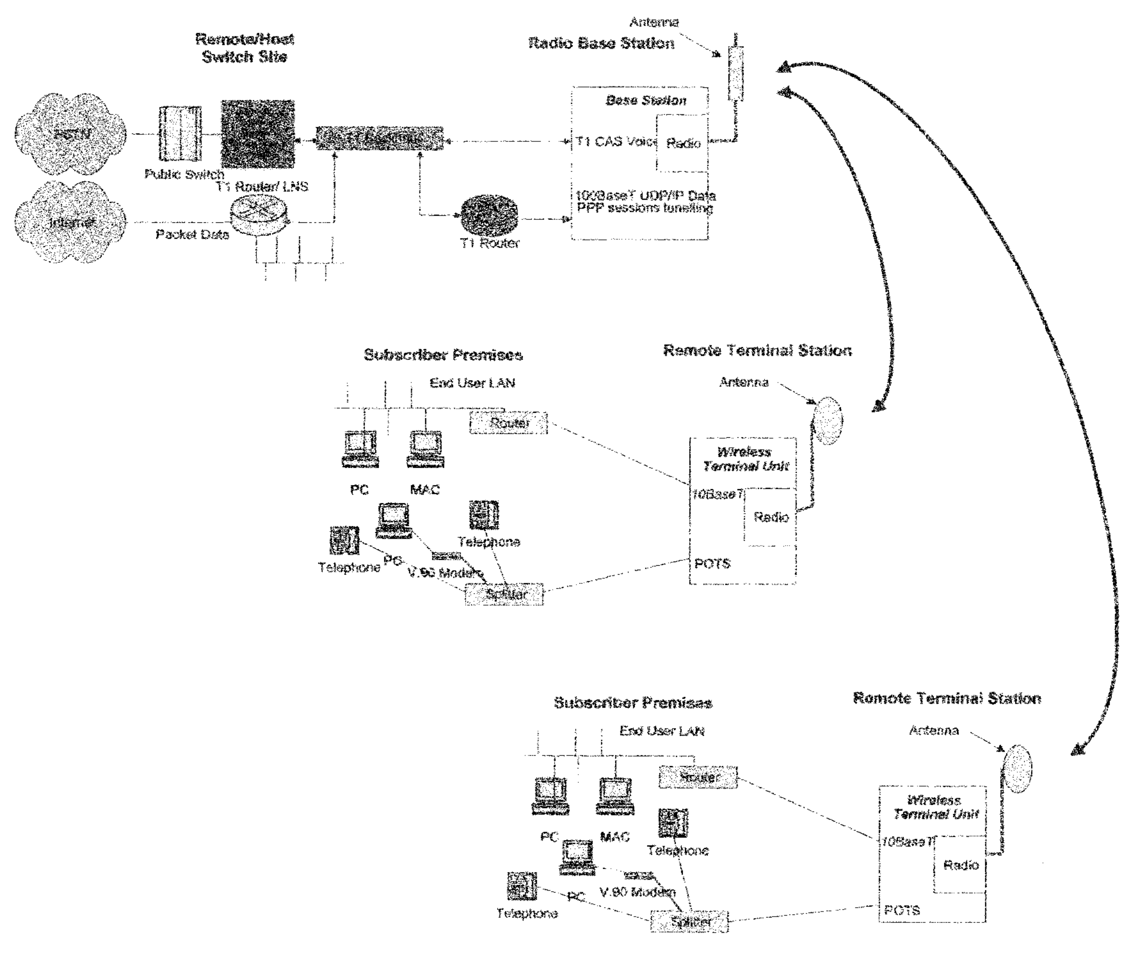

Figure 17: Simple PMP-FWA Network

The figure shows only two instances of a wireless terminal and the subscriber. However, the number of Remote Terminal Stations is directly proportional to the number of users getting service from the TELCO. In a PMP FWA network, due to the high frequencies of the carrier signal in the microwave range, the transmitter and the receiver antennas have to be in the line of sight. In situations where this condition cannot be met, due to, say, a mountain or a skyscraper, it is possible to configure the network with repeaters located at a strategic location, for example on a top of a mountain, between the Radio Base Station and the Remote Terminal Station. The obvious drawback of such an approach for the TELCO is the additional initial cost of the repeater, as well as its continuing operating expenses. For the subscriber, the additional hop introduces delays in the packet stream.

It is possible to have a number of repeaters in series. However, such configurations happen only in very sparsely populated areas. The figure below shows an example of a configuration with one repeater and a remote outstation behind it. In practice, one repeater can usually handle more than one remote outstation.

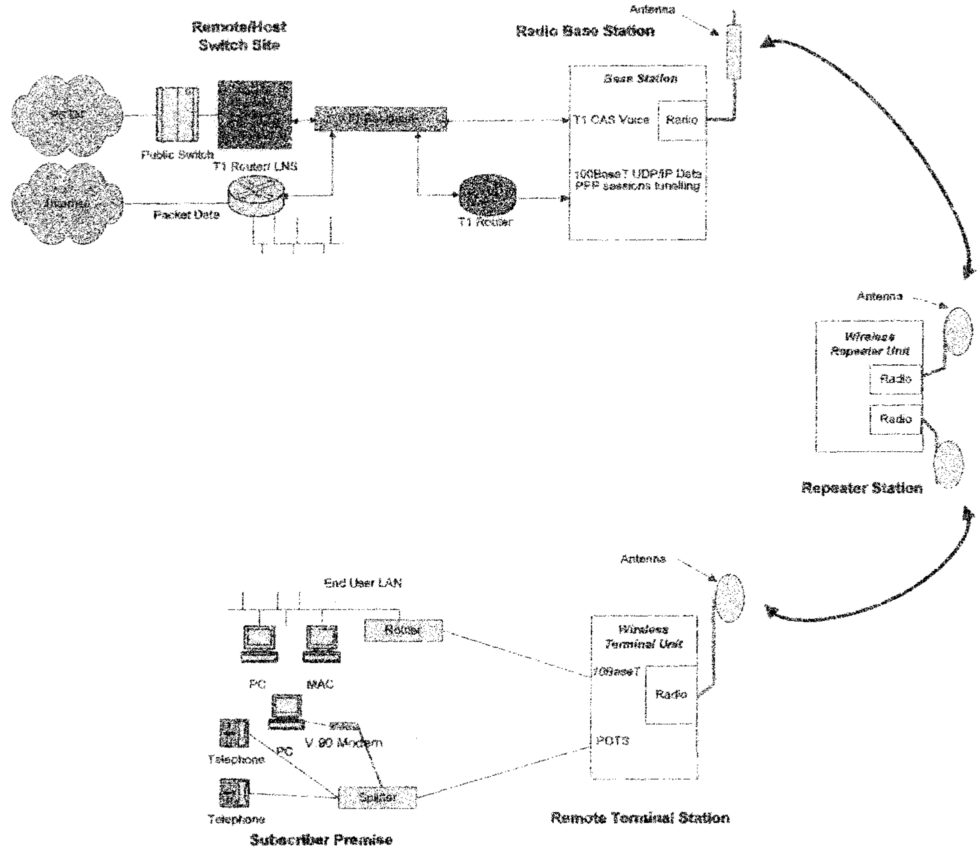

Figure 18: PMP-FWA Network with a Repeater

Typically, the PMP FWA networks do not have loops in their configurations, and they are configured in the topology of a star. The figure below gives a few examples of most common configurations. As it was mentioned previously, the number of repeaters and the number of customers attached to each outstation might vary greatly depending on the density of the population and the geographical properties of the region.

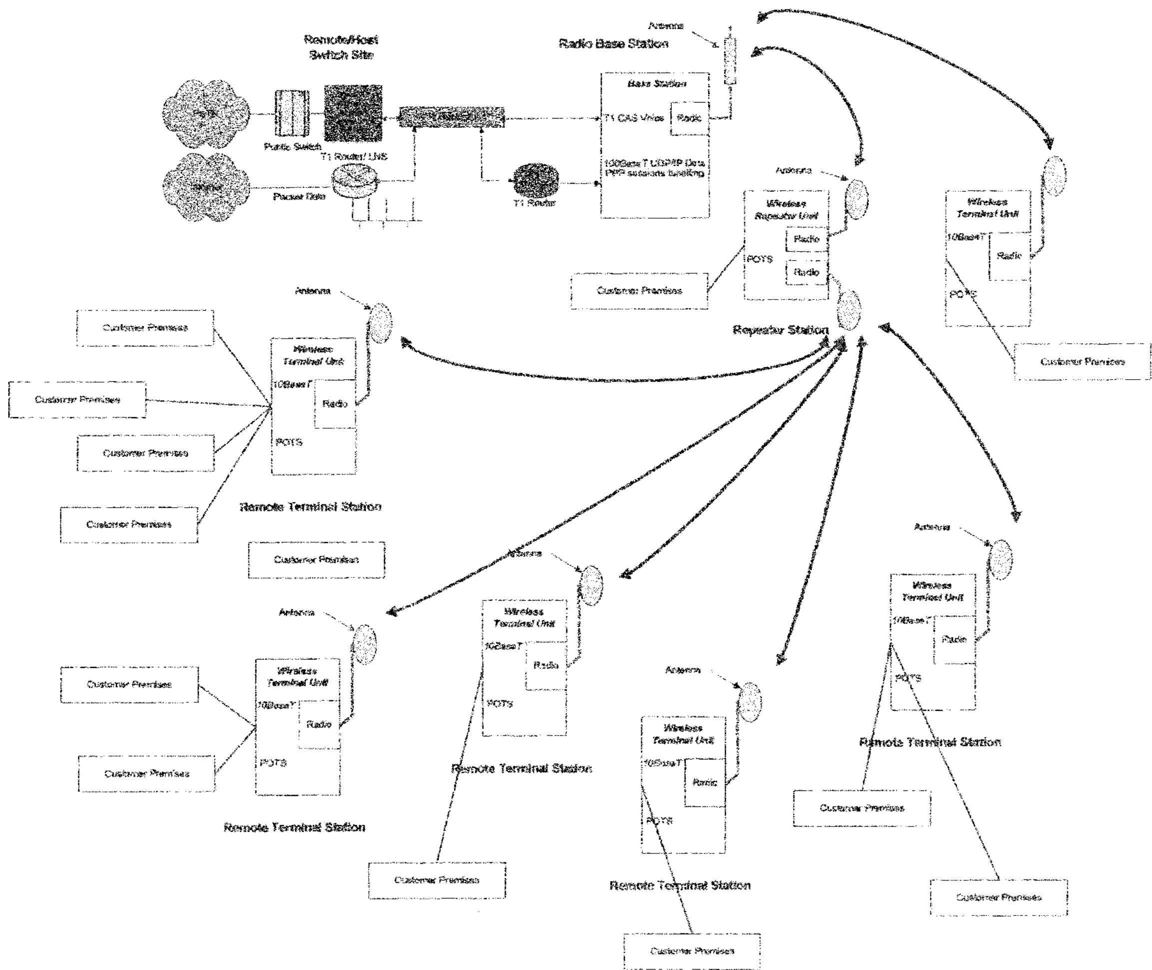

Figure 19: PMP-FWA Network

7.1 MAC Layer

Due to the lack of standards for FWA PMP networks suitable for voice applications, most of the deployed systems are based on proprietary technologies that are not interoperable between different vendors. Only recently, in spring 2002, the IEEE has published standard IEEE802.16 [IEEE802.16] which addresses the "first-mile/last-mile" connection in fixed wireless networks suitable for systems that require stringent real-time

QoS characteristics. The MAC of IEEE802.16 is based on TDMA (Time Division Multiple Access) overlaid on FDMA (Frequency Division Multiple Access). In FDMA, frequency band is dedicated to a given station all the time and the transmitted signal spectral component must be confined to the allocated frequency band. Otherwise, it will cause interference with adjacent channels [SAH94]. In TDMA, the frequency band is subdivided into channels that occupy only a small time slice of the signal.

The IEEE802.16 standard covers two ranges of frequencies. The original IEEE802.16 covers use of bandwidth between 10GHz and 66GHz, and the IEEE802.16a covers 2GHz to 11GHz range. IEEE802.16a was published in spring 2003.

The FWA PMP network studied in this thesis is based on a more mature proprietary technology developed independently by the vendors of the FWA PMP systems. It is very similar to the IEEE802.16. It is based on a FDMA scheme, where each station is permanently allocated a channel of the frequency spectrum. The framing structure for each channel consists of 8 framing bits, followed by 2 signaling and maintenance bits, followed by 264 bits of user data. The whole frame is repeated every 1ms, giving the bandwidth of 32kbs per channel. Since each channel is simplex, point-to-point, there are no collisions on the medium. Multiple channels can be aggregated for increasing the bandwidth. The number of channels available for aggregation is usually determined by the network planners configuring the deployment of the system.

7.2 Requirements for Voice Communication

Voice telephony, at the normal rate of 64kbps, is an example wherein the rate must be supported otherwise the signal will suffer so much loss as to render it unintelligible, and therefore meaningless. This is why voice has traditionally been carried by technologies that are connection-oriented. The toll-quality [KT01] real-time voice communication limits the maximum tolerable round-trip delay to 200-300ms; that is, one-way delay must be in the range of 100-150ms. The jitter should be no greater than 50ms so that the playback at the receiver is continuous. Since the packet loss in the packet based network is correlated, if packet loss were to occur, the number of contiguous packets which are lost is usually larger than one. Hence the duration of the corresponding portion of voice bit-stream that is lost can easily exceed 60ms even when a smaller packet formation time is used. Subjective testing shows that packet losses that exceed 60ms affect the intelligibility of the received speech. For this reason, for toll-quality communication, the tolerable packet loss must be set to 0.00001 to ensure that such clips occur infrequently. With silence suppression, the average periods of talk-spurt and silence are 1.65 and 1.35 seconds respectively.

7.3 Configuration of the Simulation System 1

The first system simulated in this experiment has three repeaters and six outstations connected to one router at the base station. Below is a diagram showing the configuration of the simulation system. The repeater is a key component in this model, since it is the main source of the delays in the system. Each repeater acts as signal regenerator, at the same time routing packets downstream to the customer premises. Since the outgoing traffic is transmitted on a channel with a smaller capacity than the incoming traffic, during the periods of burstiness some of the traffic has to be queued, introducing delays in the system.

This configuration is most common among the operators that provide service to areas that are far away from the telephone exchange, for example a small remote village. In such a configuration, the majority of the users are far away from the telephone exchange (in the village), with a few occasional users attached between the service provider and the major concentration of the users.

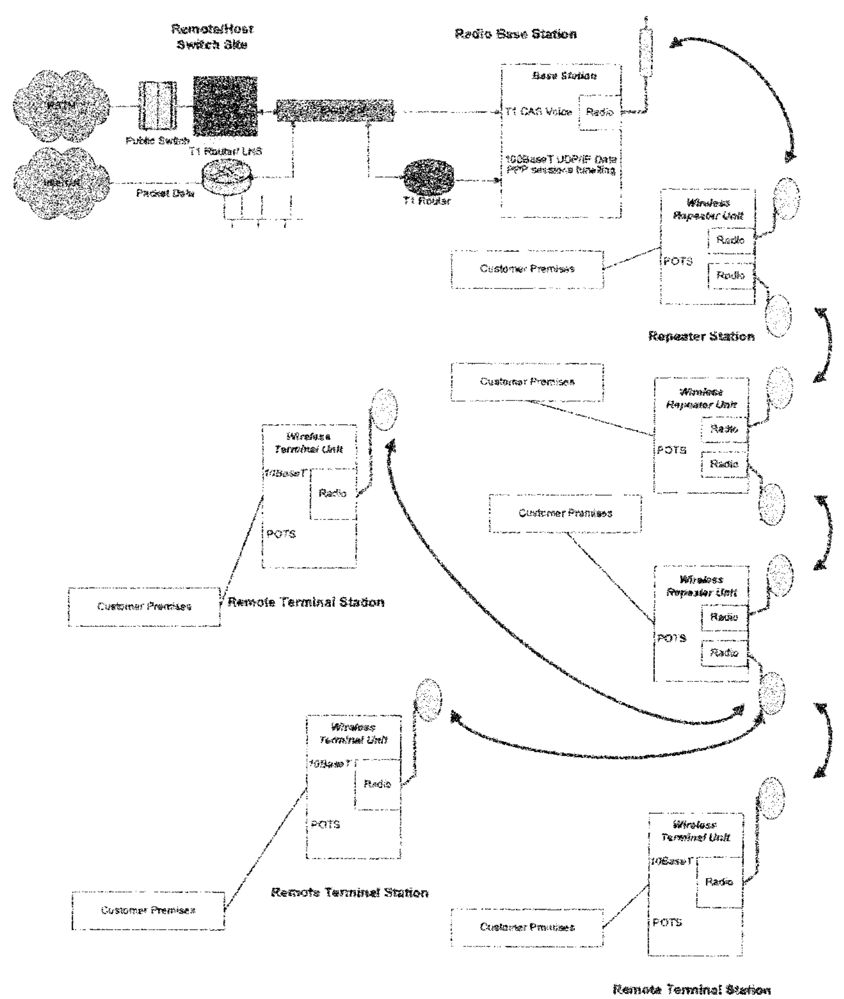

Figure 20: PMP-FWA Simulation System

The figure below shows a simplified diagram of the simulated system.

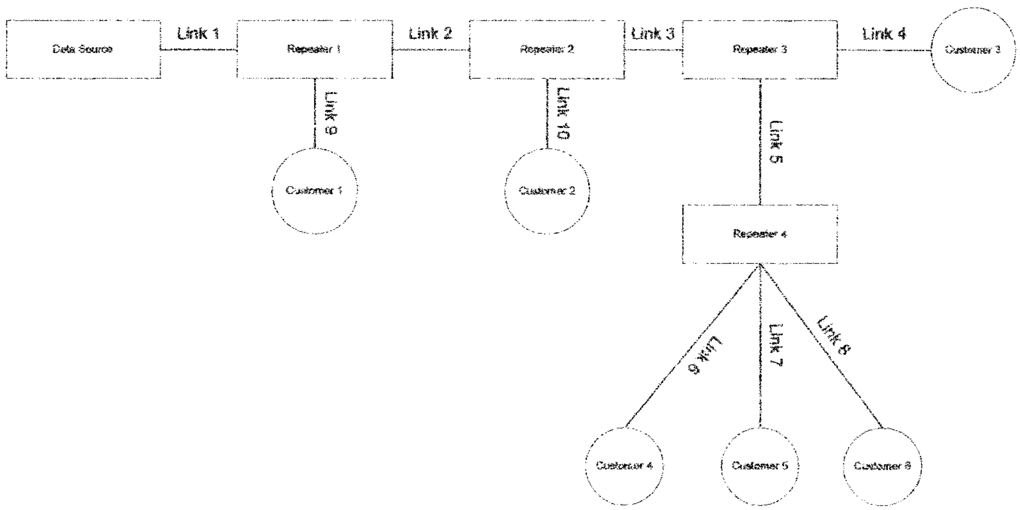

Figure 21: Simplified PMP-FWA Simulation System 1

Each customer in the simulation is receiving traffic from the data source in the upstream direction only. This is because the simulation channels, just like the real life counterparts, are unidirectional. The additional downstream traffic generated by the customers on the network would increase the complexity of the simulation without contributing to the results.

Link ID	Link Capacity
1	5
2	4
3	3
4	1
5	2
6,7,8	1
9	1
10	1

Table 2: Link Capacity of the Simulation System

Due to a limited number of channels in the licensed radio spectrum, a full blown FWA system can be configured with only a limited number of radio channels. Since all the users in the system must go through the facilities of the service provider, the amount of traffic in the system increases near the premises of the service provider. Consequently, the majority of the radio channels are assigned for the usage near the premises of the service provider. In a real-world deployment of the system, it would not make sense to allocate only one channel on link 1, because it would generate congestion most of the time.

For the purpose of this experiment, the number of channels has been arbitrarily set to 10. These channels can be aggregated at the points of the system which are the most likely to become the bottlenecks. In the system simulated in this experiment, links number 1, 2, and 3 are the most likely to impact the performance of the system. Consequently, the capacity of these links has been increased by aggregating the number of radio channels. The table 2 above shows the partition of the available channels among all the links in the system.

7.4 Configuration of the Simulation System 2

The second system simulated in this experiment has the same number of network elements as the first system. However, the topology is the inverse of the simulation system 1. The majority of the customers in this system are located near the base station. The distribution of the link capacities is identical to the one presented in the simulation system 1. Such a configuration of the system is found most likely in places where users are very near the service provider. It is deployed very often by CLECs that provide services to businesses in highly populated areas.

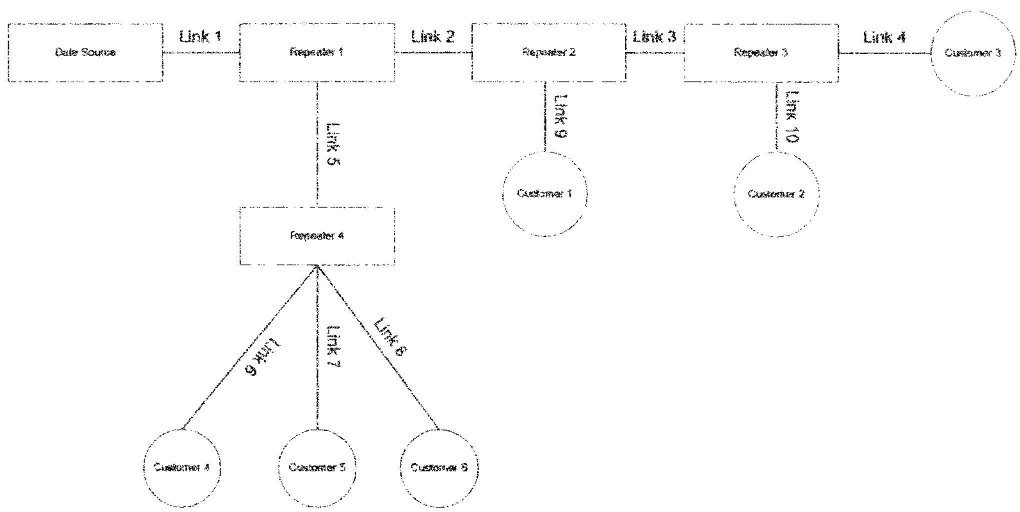

Figure 22: Simplified PMP-FWA Simulation System 2

7.5 Simulation Input and Results Generation

Each customer in the simulated system is receiving a single traffic stream simulated as an ON-OFF source [PS00] with the probabilities of going from talk-spurt to silence state (and vice versa) computed directly from the average periods of talk-spurt and silence given in section 5.1. Hence, if the speech stream is segmented into 8ms frames, packets will be generated at a rate of 125 packets/sec. Thus the mean number of packets produced in an active state is

$$E[on] = 1.65 \times 125 = 206.25$$

and the mean number of packets that are not sent in a silent state is

$$E[off] = 1.35 \times 125 = 168.75$$

From this, we can calculate the probabilities of entering the ON and the OFF states (the parameters a and s)

$$a = 1 - \frac{1}{206.25} = 0.994$$

$$s = 1 - \frac{1}{168.75} = 0.995$$

The bandwidth required for the establishment and the refreshing of the soft-state at each of the nodes for each voice stream is assumed to be negligible when compared to the bandwidth required for a voice conversation. Therefore, for the purpose of this experiment, it is ignored.

The simulation has been implemented as a discrete event simulation for a network of queues, as described by Molloy [Molloy89]. At each system tick, the main loop of the simulation schedules the packets for transmission on each of the link in the system. The process is repeated for 100,000 ticks, for three different scheduling algorithms; FIFO, Strict-Priority, and Lead-Time. In the Strict-Priority algorithm, the priority of the stream is directly proportional to the number of hops that the traffic has to go through. At each system tick, the size of each queue in the system is recorded in a log file, along with the total trip time for each packet that has been routed through the system. The post-simulation data-analysis includes the maximum trip time, average trip time, and variance of the trip time.

8 Simulation Results

8.1 Simulation System 1

8.1.1 Packet Trip-Time

The tables below show the average, maximum, and the variance of the trip time for each data stream in the network.

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	1.023	2	0.023	0.00
Customer 2	2.048	4	0.046	0.00
Customer 3	3.129	6	0.119	0.00
Customer 4	5.682	12	3.172	1.37
Customer 5	5.374	11	3.034	1.51
Customer 6	5.515	12	3.091	1.24

Table 3: FIFO Packet Trip Time Statistics

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	1.147	6	0.169	0.00
Customer 2	2.292	9	0.382	0.00
Customer 3	3.714	10	1.248	0.56
Customer 4	5.672	12	3.002	1.33
Customer 5	5.377	11	2.972	1.19
Customer 6	5.509	11	2.939	1.10

Table 4: Strict Priority Packet Trip Time Statistics

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	1.132	4	0.126	0.00
Customer 2	2.252	5	0.247	0.00
Customer 3	3.602	8	0.701	0.00
Customer 4	5.678	11	3.313	1.20
Customer 5	5.708	12	3.272	1.17
Customer 6	5.689	11	3.239	1.18

Table 5: Lead Time Packet Trip Time Statistics

Comparing the results for different scheduling algorithms we can see that for this particular configuration of the network, the Strict Priority Scheduling spreads the variance across the network most evenly. The same observation can be made about the average packet trip time; the average packet trip time for the destinations further away from the source tends to be smaller than for other types of schedulers at the expense of the packet trip time for the destinations that are closer to the source.

The Lead-time scheduling algorithm, although it behaves similarly to the Strict Priority scheduler, performs worse than the Strict-priority scheduler. This result can be explained by the observation that the lead-time based priority of a packet going to a destination further away from the source can sometimes be smaller than the lead-time based priority of a packet traveling a shorter distance, causing a delay in the scheduling. This increases the trip time, and consequently the variance of the stream.

8.1.2 Queue Occupancy

The following sections are showing the impact of the scheduling disciplines on the size of the queue at each node in the system. As expected, the scheduling discipline does not have any impact on the size of the queue at any given instant in time. With the simulated configuration of the PMP-FWA network, the experiment shows that most of the queuing occurs at the nodes which are the furthest away from the source of the traffic. This explains why the variance of the trip time for the clients close to the source is negligible for all the queuing disciplines, and it increases by an order of magnitude for the clients further away from the traffic source.

8.1.2.1 FIFO Scheduling

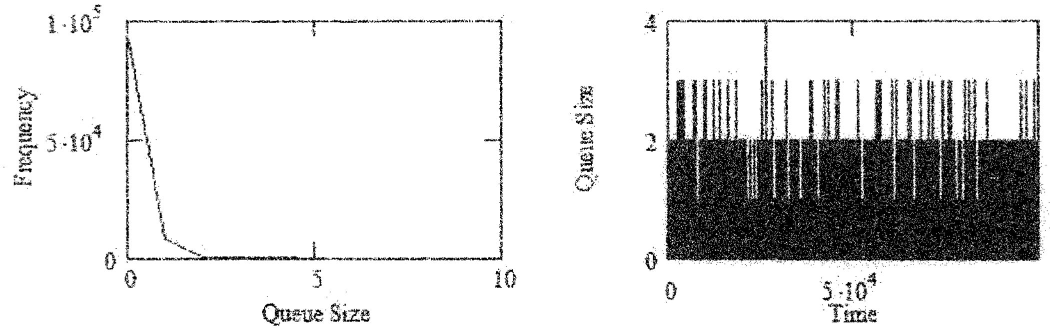

Figure 23: FIFO; Queue Occupancy for Node 1

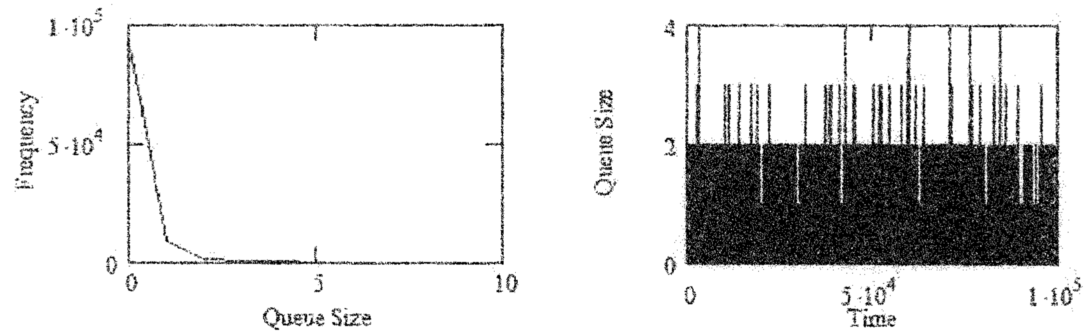

Figure 24: FIFO: Queue Occupancy for Node 2

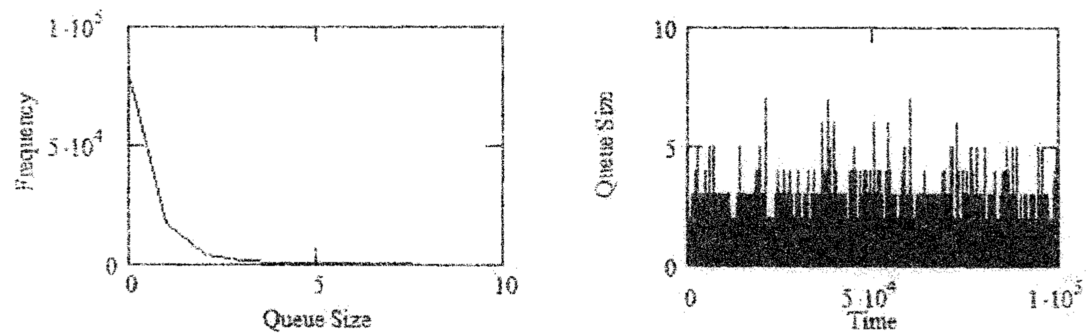

Figure 25: FIFO: Queue Occupancy for Node 3

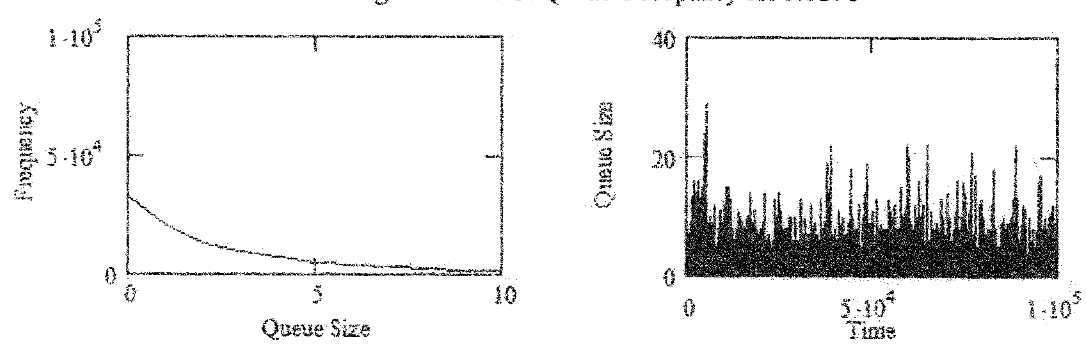

Figure 26: FIFO: Queue Occupancy for Node 4

8.1.2.2 Strict-Priority Scheduling

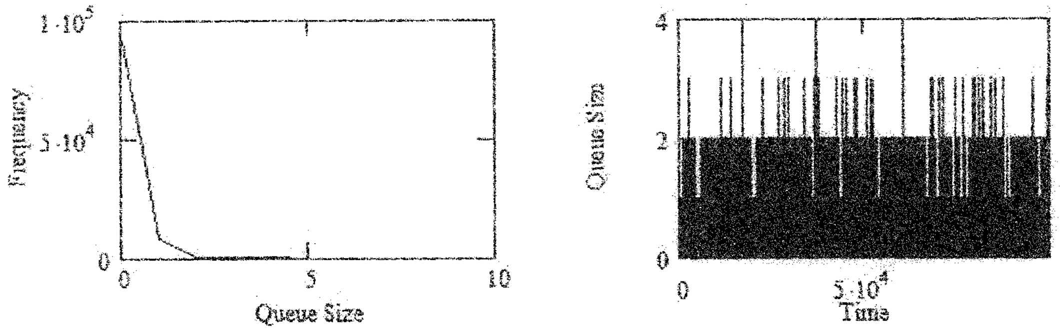

Figure 27: Strict-Priority: Queue Occupancy for Node 1

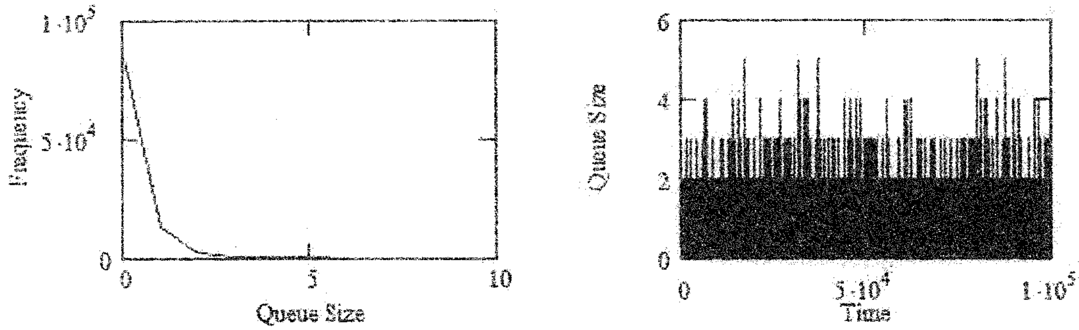

Figure 28: Strict-Priority: Queue Occupancy for Node 2

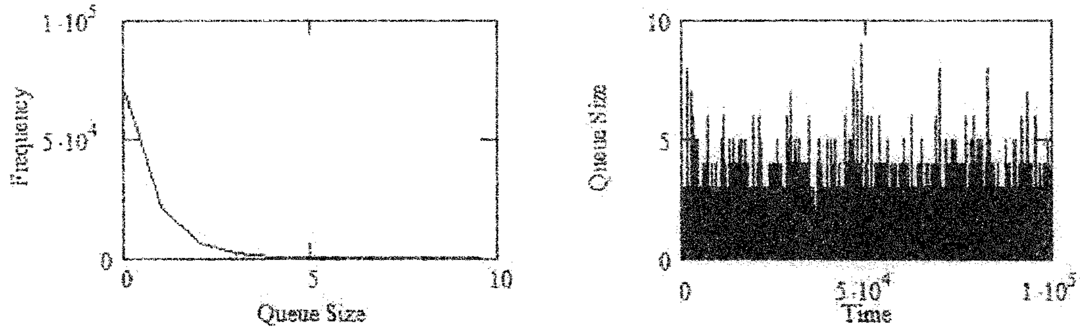

Figure 29: Strict-Priority: Queue-Occupancy for Node 3

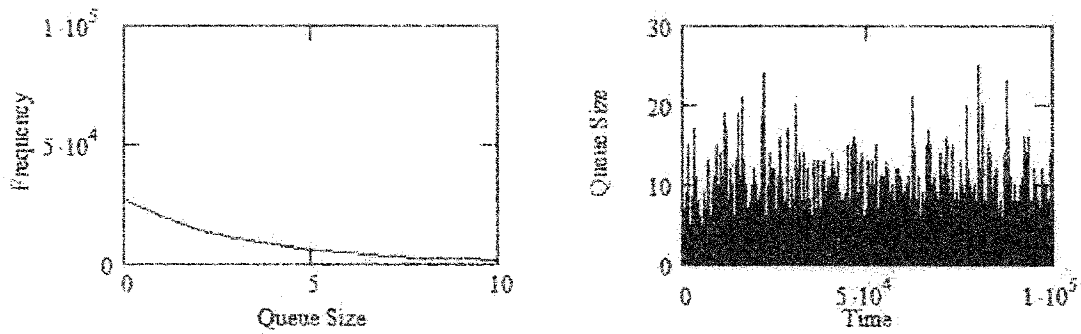

Figure 30: Strict-Priority: Queue Occupancy For Node 4

8.1.2.3 Lead-Time Scheduling

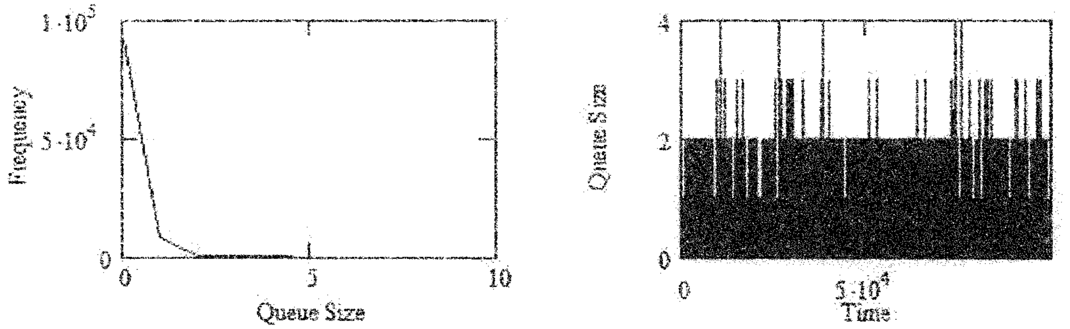

Figure 31: Lead-Time: Queue Occupancy for Node 1

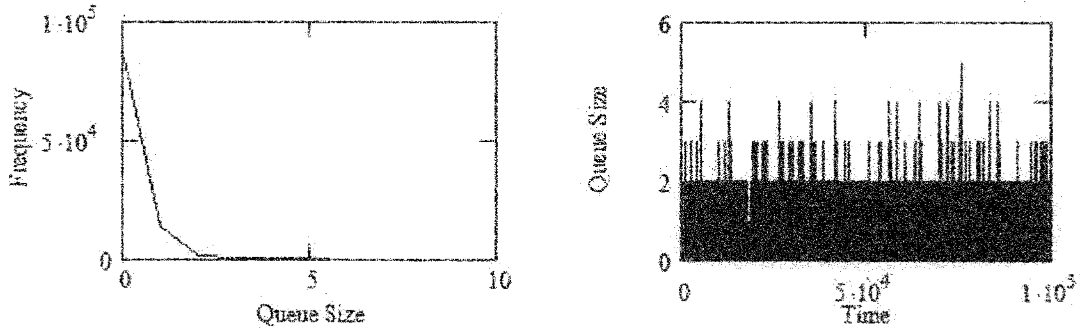

Figure 32: Lead-Time: Queue Occupancy for Node 2

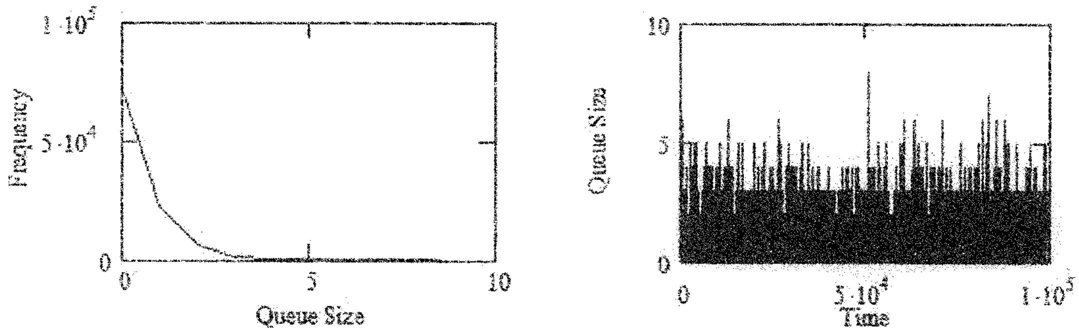

Figure 33: Lead-Time: Queue Occupancy for Node 3

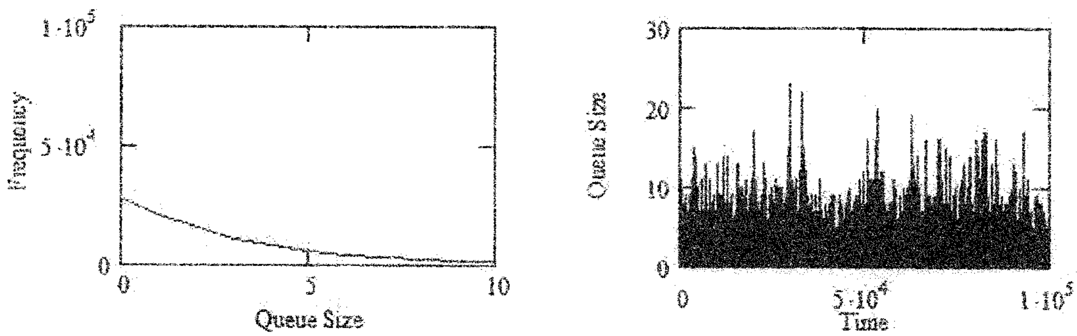

Figure 34: Lead-Time: Queue Occupancy for Node 4

8.2 Simulation System 2

8.2.1 Packet Trip-Time

The tables below show the average, maximum, and the variance of the trip time for each data stream in the network.

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	2.097	3	0.046	0.00
Customer 2	3.049	4	0.041	0.00
Customer 3	3.033	4	0.047	0.00
Customer 4	2.587	4	0.899	0.00
Customer 5	2.610	5	0.901	0.00
Customer 6	2.602	4	0.880	0.00

Table 6: FIFO Packet Trip Time Statistics

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	2.243	5	0.081	0.00
Customer 2	3.013	4	0.023	0.00
Customer 3	3.007	4	0.020	0.00
Customer 4	2.722	6	0.701	0.00
Customer 5	2.710	7	0.727	0.00
Customer 6	2.735	6	0.692	0.00

Table 7: Strict Priority Packet Trip Time Statistics

	Average trip time	Max trip time	Trip time variance	Packet loss rate (%)
Customer 1	2.204	5	0.062	0.00
Customer 2	3.028	4	0.029	0.00
Customer 3	3.021	4	0.034	0.00
Customer 4	2.671	6	0.791	0.00
Customer 5	2.653	7	0.782	0.00
Customer 6	2.689	7	0.810	0.00

Table 8: Lead Time Packet Trip Time Statistics

For this particular configuration of the network, the Strict Priority Scheduling spreads the variance across the network most evenly as well. The same observation can be made about the average packet trip time. We are not observing the packet loss because in the simulation system 2 we do not have as much queuing as in the simulation system 1. In simulation system 2, the traffic gets queued only at the routers 1 and 4 (near the source). At other routers, there is enough bandwidth to forward the traffic without queuing.

Just like in the simulation system 1, the lead-time scheduling algorithm performs better than FIFO, but worse than the strict-priority scheduler.

8.2.2 Queue Occupancy

In this particular configuration of the PMP-FWA network, we do not have any queuing at routers 2 and 3. Most of the queuing occurs in the system at node 1, and then at router 4.

8.2.2.1 FIFO Scheduling

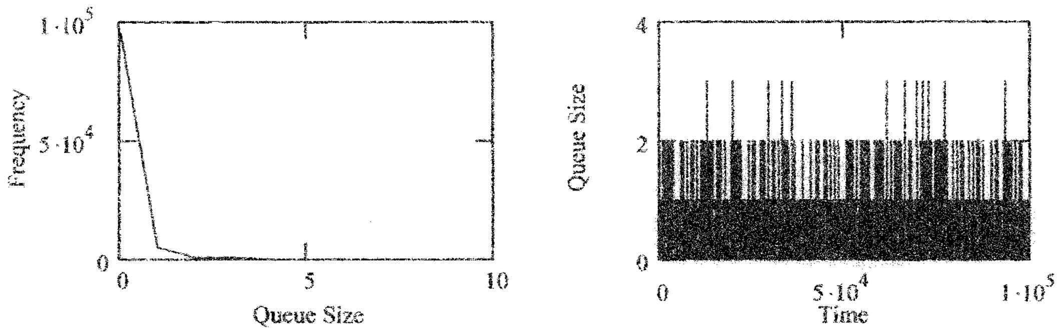

Figure 35: FIFO; Queue Occupancy for Node 1

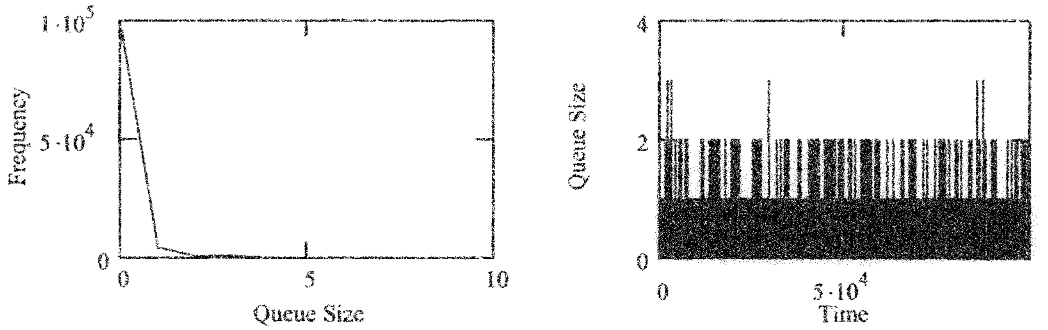

Figure 36: FIFO: Queue Occupancy for Node 4

8.2.2.2 Strict-Priority Scheduling

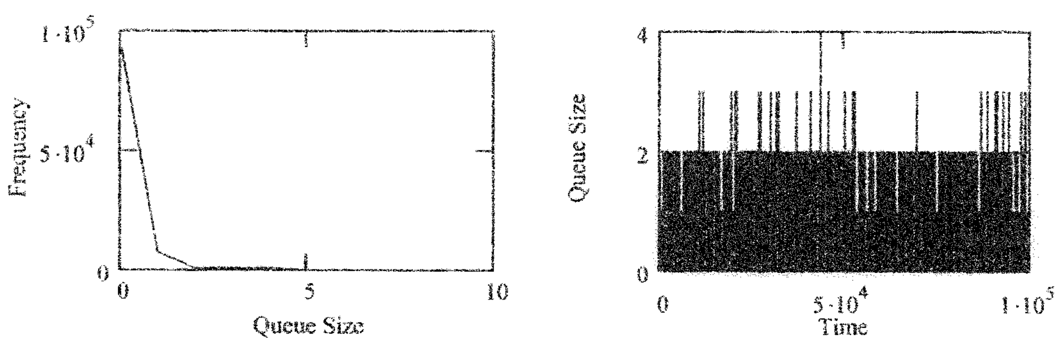

Figure 37: Strict-Priority: Queue Occupancy for Node 1

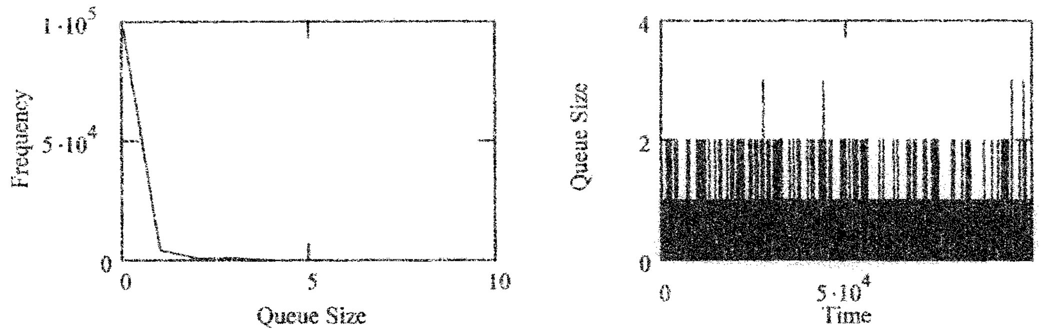

Figure 38: Strict-Priority; Queue Occupancy for Node 4

8.2.2.3 Lead-Time Scheduling

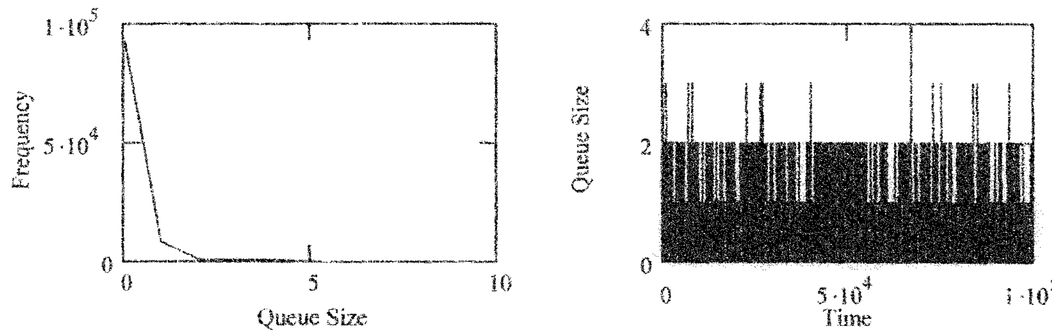

Figure 39: Lead-Time; Queue Occupancy for Node 1

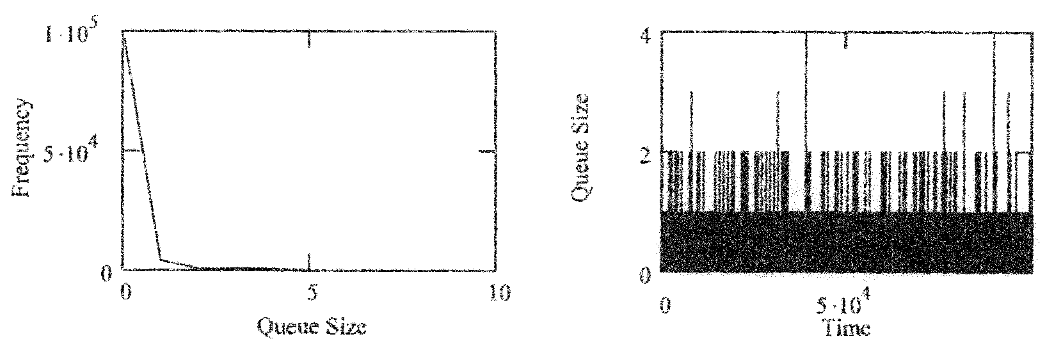

Figure 40: Lead-Time; Queue Occupancy for Node 4

9 Conclusion

This thesis explored the Lead-time scheduling algorithm in the context of a packet-based PMP-FWA network. The first part of the thesis gave the current state of the processing of the real-time traffic, and the technological advances that were accomplished during the last few years to accommodate this new type of payload on the existing infrastructure of the Internet.

The second objective of this thesis was to evaluate the performance of a lead-time scheduler in the context of a packet based access network for real-time voice. The simulation has shown that in a configuration where most of the queuing occurs close to the destination, the scheduler does not show as many benefits as hoped. Although it performs better than the FIFO scheduler, it performs worse than the Strict-Priority scheduler.

9.1 Current Limitations and Future Work

This thesis has presented the lead-time scheduling algorithm in the context of the FWA-PMP network. Future work might study the algorithm in different applications, such as cellular or satellite networks.

Bibliography

[ACH98] C. Aurrecoechea, A. T. Campbell, L. Hauw, "A Survey of QoS Architectures", Multimedia Systems Volume 6, Number 3, pp 138-151. 1999. http://citeseer.nj.nec.com/aurrecoechea96survey.html

[AKRS94] C. Aras, J. Kurose, D. S., Reeves, H. Schulzrinne, "Real-Time Communication in Packet-Switched Networks", IEEE Proceedings, vol. 82, pp 122-139, January 1994.

[Adas96] A. Adas, "Traffic Models in Broadband Networks", IEEE Communications Magazine, 35(7), pp82-89, July 1997.

[Armitage00] G. Armitage, "Quality of Service in IP Networks; Foundations for a Multi-Service Internet", Published by MTP (Macmillan Technical Publishing), 2000.

[BMK96] S. Basu, A. Mukherjee, S. Klivansky, "Time Series Models for Internet Traffic", INFOCOM 1996, pp 611-620.

[BPSK96] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", Proceedings of the SCM SIGCOMM 1996. August 1996, pp 256-267.

[CB96] M.E. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", Performance Evaluation Review, May 1996.

[CD97] K. Chen, L. Decreusefond, "An approximate analysis of Waiting Time in Multi-classes Queues", January 30, 1997. http://citeseer.nj.nec.com/184810.html

[CSZ92] D. D. Clark, S. Shenker, L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", SIGCOMM 1992, pp 14-26.

[DL86] J.N. Daigle, J.D. Langford, "Models for Analysis of Packet Voice Communication Systems", IEEE JSAC Vol. 6, pp 847-855, 1986.

[DLS00] B. Doytchinov, J. Lehoczky, S. Shreve, "Real-Time Queues in Heavy Traffic with Earliest-Deadline-First Queue Discipline", Annals of Applied Probability, 11, (2001), pp 332-379.

[FJ95] S. Floyd, V. Jacobson," Link-sharing and resource management models for packet networks", IEEE/ACM Transactions on Networking, Volume 3, Number 4, pp 365-386, 1995.

[FV90] D. Ferrari, D.C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks", IEEE Journal on Selected Areas in Communications, Volume 8, Number 3, pp 368-379, 1990.

[Ferrari90] D. Ferrari, "Client Requirements for Real-Time Communication Services", IEEE Communications Magazine, 28(11), pp 65-72, November 1990.

[Mahanta97] G. Mahanta, "Video Transmission Over Constrained Bandwidth Channels", Master Thesis in The Department of Computer Science, Concordia University, 1997.

[Karn93] P. Karn, "The Qualcomm CDMA digital cellular system", Proceedings of the USENIX Mobile and Location-Independent Computing Symposium, August 1993, pp 35-39.

[KAZ98] S.V. Krishnamurthy, A.S. Acampora, M. Zorzi, "On the Capacity of TDMA and CDMA for broadband Wireless Packet Access", PIMRC'98, Boston, September 1998.

[KBSSGM98] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, J. Mahler, "Real-Time Voice Over Packet-Switched Networks", IEEE Network, 12(1), pp 18-27, January/February 1998.

[KLSY00] L. Kruk, J. Lehoczy, S. Shreve, S. N. Yeung, "Multiple-Input Heavy-Traffic Real-Time Queues", October 2000. http://citeseer.nj.nec.com/kruk00multipleinput.html

[KLSY02] L. Kruk, J. Lehoczy, S. Shreve, S. N. Yeung, "Earliest-Deadline-First Service in Heavy-Traffic Acyclic Networks", July 22, 2002. http://citeseer.nj.nec.com/533295.html

[KSF91] D. D. Kandlur, K. G. Shin, D. Ferrari, "Real-Time Communication in Multi-Hop Networks", Proceedings of the 11th International Conference on Distributed Computing Systems (ICDCS), IEEE, pp 300-307, 1991.

[KT01] M. Karam, F. A. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic Over the Internet", INFOCOM 2001, pp 824-833.

[LTWW93] W.E. Leland, M.S. Qaqqu, W. Willinger, D.V.Wilson,"On the Self-Similar Nature of Ethernet Traffic", ACM SIGCOMM, pp 183-193, 1993.

[LT01] C. Lindemann, A. Thummeler, "Performance Analysis of the General Packet Radio service". ICDCS 2001, p 673.

[LT01a] C. Lindermann, A. Thummeler, "Evaluating the GPRS Radio Interface for Different Quality of Service Profiles", "Kommunikation in Verteilten Systemen", pp291-301, 2001.

[Lin01] C. R. Lin, "An On-demand QoS Routing Protocol for Mobile Ad Hoc Networks", INFOCOM, pp 1735-1744, 2001. http:// citeseer.nj.nec.com/lin01demand.html

[Molloy89] M. K. Molloy, "Fundamentals of Performance Modelling", Macmillan Publishing Company, 1989.

[PF95] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", IEEE/ACM Transaction on Networking, 3(3), pp 226-244, June 1995.

[PS00] J. M. Pitts, J. A. Schormans, "Introduction to IP and ATM Design and Performance", Second Edition, John Wiley & Sons Ltd, 2000.

[PX98] G. C. Polyzon, G. Xylomenos, "Enhancing Wireless Internet Links for Multimedia Services", Proc. of International Workshop on Mobile Multimedia Communications (MoMuc), Berlin, Germany, Oct. 1998.

[RFC791] J. Postel, "Internet Protocol", IETF RFC791.

[RFC793] J. Postel, "Transmission Control Protocol", IETF RFC793.

[RFC3209] D. Awduche, L.Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", IETF RFC3209.

[SAH94] T. N. Saadawi, M. H. Ammar, A. El Hakeem, Fundamentals of Telecommunication Networks, John Wiley & Sons 1994.

[SV95] M. Sreedhar, G. Varghese, "Efficient Fair Queing Using Deficit Round Robin", SIGCOMM 1995, pp 231-242, 1995.

[VZF91] D. C. Verma, H. Zhang, D. Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network", TR-91-007, Berkley, CA, 1991.

[XL95] G.G. Xie, S.S. Lam, "Delay Guarantee of Virtual Clock Server", IEEE/ACM Transactions on Networking", Volume 3, Number 6, 1995, pp 683-689.

[YLL95] Yibin Yang, Ten-Hwang Lai, Ming-Tsan Liu, "Real-Time Communications in FDDI-Based Mobile Networks", 1995. http://citeseer.nj.nec.com/371759.html

[Zhang95] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks", Proc. IEEE, 83(10): pp 1374-1396, Oct. 1995.