Multimedia Networks with Deterministic Quality-of-Service Guarantees

A Dissertation

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Computer Science

by

Dallas E. Wrege

August 1996

Approvals

This dissertation is submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Computer Science

Dallas E. Wrege

Approved:

Jörg Liebeherr (Advisor)

William A. Wulf

Gabriel Robins

Alfred C. Weaver (Chair)

Stephen G. Strickland
(Minor Representative)

Accepted by the School of Engineering and Applied Science:

Richard W. Miksad (Dean)

August 1996

Abstract

Future integrated-services networks are expected to support applications with a wide range of service requirements. The most demanding applications require a bounded-delay service that provides deterministic (i.e., worst-case) bounds on network latencies for all packets. To provide such delay guarantees, a network must allocate network resources such as bandwidth and buffer space to individual connections. However, since resource availability is limited, the network must carefully manage its resources in order to ensure a high achievable network utilization.

Two components are central to the design of a network with a bounded-delay service: the *traffic characterization method* used to specify traffic on a connection, and the *packet scheduling discipline* at network switches that determines the transmission order of packets queued at output buffers. The choice of each component determines a tradeoff between achievable network utilization and implementation overhead. In particular, a traffic characterization should accurately describe the actual arrivals on a connection so that a large number of connections can be supported, but it must also conform to a simple parameterized model that the network can easily monitor and enforce. Similarly, a packet scheduling discipline should be sophisticated enough to support tight delay constraints at high loads, but it must also have a simple implementation so that packets can be processed at the speed of the transmission link.

This dissertation presents novel methods for traffic characterization and packet scheduling that are practical for implementation yet yield a high network utilization in a bounded-

delay service. The first problem considered is the characterization of compressed digital video, an important traffic type that is both high in bandwidth usage and has a variable bit rate (VBR). A novel traffic characterization method is presented that is shown to accurately specify the traffic of VBR video sources. This characterization method is based on approximating an optimal traffic characterization that is itself impractical because of two significant drawbacks: (1) the need for a large number of parameters that are computationally expensive to produce, and (2) the inability of the network to monitor or enforce the optimal traffic characterization. Our characterization method addresses both of these problems by determining a subset of its parameters that can be computed quickly and using these parameters to determine a traffic characterization that can be easily enforced by simple traffic policing mechanisms.

A novel packet scheduling discipline, referred to as Rotating-Priority-Queues$^+$ (RPQ$^+$), is developed that is near-optimal in the sense that it can approximate the optimal Earliest-Deadline-First (EDF) discipline with arbitrary precision. The advantage of the RPQ$^+$ scheduler over EDF is in its computational overhead: while an implementation of EDF requires the sorting of packets which is impractical for high-speed networks, RPQ$^+$ avoids the need for sorting by using a set of prioritized FIFO queues whose priorities are rearranged (*rotated*) periodically to increase the priority of waiting packets. For shared-memory architectures, it is demonstrated that RPQ$^+$ can be implemented with little computational overhead. Analysis of the RPQ$^+$ scheduler shows that it has the following desirable properties: its implementation requires operations independent of the number of queued packets; it can provide worst-case delay guarantees; it is always superior to a *Static-Priority* (SP) scheduler; and its achievable network utilization increases with the frequency of queue rotations, approaching that of EDF in the limit.

To my Parents, my Brother, and Katie

Contents

Acknowledgements

First of all, I would like to thank my advisor, Jörg Liebeherr, who helped me rise to face many challenges over the last three years. I will always remember fondly the four-hour marathon sessions where we cultivated the ideas for this dissertation (and enough ideas for several others). Jörg always encouraged me and helped me stay focused during the more difficult times. His insights, comments, ideas, and criticisms can be clearly seen throughout my dissertation.

Thanks go to Bill Wulf, Gabriel Robins, and Steve Strickland who served on my thesis committee and provided valuable comments on my work. I am especially grateful to Alf Weaver, my committee chairman, who has been a continual source of encouragement.

Being a member of the Multimedia Networks Group has been a highlight of my years at Virginia. I enjoyed the many "deep thoughts" thinking sessions I shared with Anastasios Stamoulis. Brian Hope and Dave Bassett were more than willing to listen to my ideas. Thanks also go to the many people who have shared Room 235 with me over the years: Lance Hoppenwasser, Laura O'Brien, Sudhir Srinivasan, Mary Ann Stumbaugh, Alan Tai, and, last but not least, my stern but kindhearted office czar, Emily West.

Whenever my zest for research was lagging, the "lifting crew" of John Karpovich, Charlie Viles, and Matt Lucas was always quick to get me to the bench press. John and I have shared many a laugh over burritos at Guadalajara. Charlie has been a devoted friend, and his wisdom has helped me keep things in perspective. (The enormous quantity of uncooked cookie dough we have eaten together is another important and sacred bond we share.) Special thanks go to my good friend Matt for contributing to my research in so many important ways. He has made invaluable suggestions that improved the quality of my work, and he has served as a sounding board for many of my ideas.

An enormous debt of gratitude is owed to my family. The love and support of my parents and brother is the solid foundation upon which I have built my life. I thank my mother and father for providing me with so many opportunities in life and always encouraging me to follow my dreams. I am grateful to my brother Shannon for always believing in me even when I myself had doubts in my abilities. I also thank my grandmother and my future in-laws, Bill and Barbara Oliver, for their encouragement and support at key times.

My last thanks go to Katie Oliver, a woman who is my fiancé as well as my best friend. I thank Katie for her sacrifice, patience, and love over four long years. Without her moral support, particularly over the last few months, this dissertation would never have been completed.

List of Figures

List of Tables

List of Symbols

A	Traffic arrival function
A^*	Traffic constraint function
A_t^*	Traffic constraint function renegotiated at time t
$B(t)$	Set of backlogged connections at time t
B^*	Traffic constraint function for (σ, ρ)-traffic model
B_m^*	Traffic constraint function for $(\vec{\sigma}, \vec{\rho})$-traffic model with m pairs
$C(B_m^*, B_n^*)$	Cost function expressing weighted difference between B_m^* and B_n^*
$\mathcal{C}$	Set of all connections with traffic at a switch
$\mathcal{C}_p$	Connection set with priority p $(\cup_{p=1}^{P} \mathcal{C}_p = \mathcal{C})$
D	End-to-end delay bound
d	Local delay bound
E^*	Empirical envelope function
E_t^*	Empirical envelope function of a sequence $A(\tau)$ for times $\tau \geq t$
F_i^k	Virtual finishing time of the kth packet of connection i
$\mathcal{H}$	Concave hull operator
I	Interval parameter in $(X_{min}, X_{ave}, I, s^{max})$-model
R_k^*	Repetition extrapolation of the first k parameters of the empirical envelope E^*
$R_{k,t}^*$	Repetition extrapolation of the first k parameters of E_t^*

R	Rate parameter in D-BIND model
$R(t)$	Remaining transmission time of packet in transmission at time t
r	Rate parameter in (r,T)-model
s	Packet transmission time
s^{max}	Maximum packet transmission time
s^{min}	Minimum packet transmission time
T	Interval parameter in (r,T)-model
$W^{p,t}(t+\tau)$	Transmission time of all packets in a scheduler at time $t+\tau$ to be transmitted before a packet from connection set $\mathcal{C}_p$ that arrived at time t
X_{min}	Minimum packet interarrival time
X_{ave}	Minimum average packet interarrival time in $(X_{min}, X_{ave}, I, s^{max})$-model
Δ	Rotation interval
δ	Sum of actual queueing and transmission delays of a packet
ϕ_i	Service share of connection i
ρ	Rate parameter in (σ,ρ)-model
σ	Burst parameter in (σ,ρ)-model

1

Introduction

The bandwidth provided by packet-switched computer networks has increased dramatically in recent years from a few megabits per second (Mbps) to hundreds or even thousands of Mbps. Although computer networks were originally designed to transport discrete media such as text and still images, the availability of higher data rates has made feasible the transmission of continuous media such as voice, video, and audio over these networks.

Continuous-media applications are distinct from discrete-media applications in that they require performance guarantees from the network. While discrete-media applications are tolerant of network latencies, continuous-media applications are sensitive to the *quality-of-service* (QoS) they receive in terms of delay, delay variation (i.e., “jitter”), and loss rate. For example, a bidirectional voice conversation requires (1) small network delays to maintain the interactive nature of the conversation and (2) small delay variation to ensure continuous playback at the receivers. Future packet-switched networks will need to be able to provide guarantees on QoS to individual applications.

A packet-switched network that provides QoS is *connection-oriented* with a *resource reservation scheme* to allocate resources such as bandwidth and buffer space for connections. A network client desiring a new connection submits to the network a specification of its traffic and the desired QoS, and the connection is only established if sufficient resources are available to ensure that the traffic on all connections can be transmitted within the specified QoS constraints. The resource reservation scheme is used to limit the number of connections as well as the traffic on the connections so that all QoS guarantees can be mathematically verified. The design of the resource reservation scheme is critical: connections will not receive their desired QoS if too few resources are allocated, while overallocating resources will result in low network utilization. However, precise resource allocation for compressed digital video traffic is especially difficult since video traffic is variable-bit-rate (VBR) with considerable burstiness. In this dissertation, we consider the design of a network with a resource reservation scheme that can support VBR video connections with diverse QoS requirements while achieving a high network utilization.

The remainder of this chapter motivates our research and is structured as follows. In Section 1.1 we describe proposals for networks that support integrated-services networks. We review the service models proposed by both the Internet Engineering Task Force (IETF) and the ATM Forum, and we observe that both communities agree about the need for a bounded-delay service that provides worst-case guarantees on network latencies. In Section 1.2 we discuss resource reservation requirements for bounded-delay services. In particular, we identify two key components of a resource reservation scheme that are to be studied in this dissertation: traffic characterization for resource allocation, and packet scheduling disciplines at network switches. In Section 1.3 we summarize our approach to the problem and point out practical challenges and tradeoffs involved in selecting both characterization methods and scheduling disciplines. We outline the dissertation in Section 1.4.

1.1 Background: Integrated Services

Historically, different types of networks have been designed for the support of various media classes: telephony networks for voice, cable networks for video, and computer networks for discrete media [70]. Conversely, emerging *multimedia networks* (also referred to as *QoS networks*) provide integrated services that support all media classes over a single network.

Multimedia networks must support a wide variety of traffic classes with different QoS requirements. At the highest level of service, which we refer to as a *deterministic* service, the network should guarantee the delivery of *all* packets within the desired QoS constraints. Such a service requires allocating sufficient resources for a connection to support its QoS guarantees even during worst-case situations, i.e., during rare periods of extreme congestion. Because of this worst-case allocation strategy, the resources allocated to a connection may significantly exceed its average resource requirements. At the lowest level of service, often called a *best-effort* service, the network does not provide any QoS guarantees at all. In a best-effort service, resources need not be reserved for traffic, allowing for maximal network utilization through statistical multiplexing. However, packets may be delayed or dropped arbitrarily. Intermediate services have been investigated that provide QoS guarantees between these two extremes, trading off QoS and network utilization. Among these service types are (i) *statistical* services that provide probabilistic bounds on the percentage of packets delivered within the QoS constraints [8, 27, 31, 54, 60, 113], (ii) services with *bounded degradation* that allow a client to specify a degradation of service commitments for a fixed portion of traffic [71], and (iii) *predictive* services that estimate the QoS of a connection based on measurements of current resource usage [13, 20, 51].

In the remainder of this section, we discuss the service classes of two important and influential service models: the so-called integrated-services Internet and the approach specified by the the ATM Forum. We observe that both of these proposed service models include a *bounded-delay service* that provides worst-case bounds on network latencies for the most demanding applications.

1.1.1 Integrated-Services Internet

The existing Internet service model provides only a best-effort datagram service that provides no guarantees on delay, delay jitter, or loss rate. Clearly, this service model is inappropriate for multimedia applications. For this reason, the IETF has proposed enhancements to the Internet protocols to provide predictable and reliable services designed for continuous media traffic [13, 20]. These enhancements include a service model that supports two connection-oriented "real-time" services in addition to the traditional best-effort service.

The service model of the integrated-services Internet is presented in [13, 20] and consists of three service classes. The first two service classes, which are referred to as *guaranteed service* and *predictive service*, involve the establishment of connections through the network (i.e., they are are connection-oriented) and employ *admission control mechanisms* to ensure that sufficient resources are available to support a connection before it is admitted. The guaranteed service class is the most stringent of the proposed classes, providing worst-case bounds on network latency for all packets. Any connection desiring guaranteed service submits to the network a specification of its maximum traffic along with the desired delay guarantee prior to connection establishment, and the admission control mechanisms use these specifications to determine if the connection is admissible, that is, if the connection can be supported with its delay constraints. Note that the guaranteed service class provides a deterministic service as described in Section 1.1. The predictive service also employs admission control mechanisms based on traffic and delay specifications, but these mechanisms are more optimistic than those for guaranteed services [51]. Admission control mechanisms for predictive services estimate traffic on existing connections based on empirical measurements instead of using the (worst-case) traffic specifications, resulting in the admission of a larger number of connections at the risk of violating delay guarantees. Predictive services are designed for applications that are tolerant to loss or have the ability to adapt their traffic rate based on congestion levels in the network. We note that two alternative intermediate services have been recently proposed in the Internet community: the *committed rate service* that guarantees a minimum rate to admitted connections without providing explicit delay guarantees [7], and the *controlled load service* that limits the number of connections to ensure that all connections receive the same best-effort service that they would receive from a lightly loaded network [106]. In addition to above services, the proposed integrated-services Internet service model also supports the *best-effort* service class as found in the Internet today.

Several protocols have been proposed for the support of integrated services, but these protocols provide mechanisms independent of the service model. Both the Stream Protocol: Version 2 (ST-II) [99] and the Resource ReSerVation Protocol (RSVP) [14, 117] are resource reservation protocols for allocating resources to individual connections. These protocols are general in that they support a general class of traffic specifications. Also of note is the Real Time Protocol (RTP) [93] that provides a packet format with timing information that can be used by applications to aid in multimedia presentation. All of these protocols provide mechanisms that are orthogonal to the problem of determining the amount of resources needed to support particular QoS guarantees.

1.1.2 Asynchronous Transfer Mode (ATM)

Traditional telephony and cable networks employ either frequency-division or time-division multiplexing to provide a fixed transmission rate for continuous media connections. By selecting a transmission rate larger than the peak traffic rate of a connection, these networks deliver all traffic with a small constant delay. Although this approach achieves high network utilization for traffic sources such as encoded voice and analog video that have a reasonably constant traffic rate, it cannot achieve high utilizations for bursty sources such as computer data or compressed video applications. For this reason, the telecommunications community has investigated ATM technology to provide more flexible services. ATM is essentially a packet-switching technology that switches small fixed-sized packets called *cells*. This approach allows for efficient multiplexing as well as the support of different QoS guarantees for different traffic types.

The ATM service model includes the following five service classes [1, 2]:

- *Constant Bit Rate* (CBR): Similar to the synchronous schemes of traditional telephony networks, the CBR service class makes available a fixed quantity of bandwidth for each connection. CBR service provides bounds on delay and delay jitter to traffic that can be characterized by its peak rate. This service class is the first of the two real-time services in the ATM service model.

- *Real-Time Variable Bit Rate* (rt-VBR): Similar to CBR, the rt-VBR service class provides tight constraints on delay and delay variation. However, rt-VBR is distinguished by its more sophisticated traffic characterization. This service is designed for soft real-time applications such as compressed video that have a variable transmission rate. Higher network utilizations can be achieved using rt-VBR rather than CBR due to multiplexing gains.

- *Non-Real-Time Variable Bit Rate* (nrt-VBR): The nrt-VBR service class provides guarantees on the average delay and maximum loss rate of a connection. This service class is not intended for real-time applications but rather for extremely bursty applications that are time-critical such as banking transactions or airline reservations.

- *Available Bit Rate* (ABR): The ABR service class enforces a bound on the minimum throughput of connections and additionally divides unused bandwidth fairly among its connections. This service class is designed for applications that can adapt their traffic rate in accordance with the changing availability of network resources.

- *Unspecified Bit Rate* (UBR): Similar to the best-effort Internet service class, the UBR service class is designed for data applications and does not provide any QoS guarantees.

Of the five ATM service classes, only UBR does not employ admission control mechanisms. Both CBR and rt-VBR are distinguished as real-time service classes, while the remaining three service classes are for non-real-time applications [1].

1.1.3 Common Ground: Bounded-Delay Services

For multimedia applications, delay is regarded as the most important QoS parameter [22, 28]. A service that guarantees a deterministic bound on delays also provides bounds on both delay jitter and throughput: the maximum jitter on a connection can be derived directly from its minimum and maximum delays and can be handled by introducing a buffer at the receiver, while the throughput of a connection is equal to the offered traffic rate of a connection.

Both the Internet and ATM service models include a service class that supports deterministic guarantees on maximum delay for connections, which we refer to as a *bounded-delay service*. Other protocol architectures that include a bounded-delay service are the Tenet scheme [8], the QoS Architecture (QoS-A) [16], and MAGNET II [66]. Thus, it is widely agreed upon that a bounded-delay service is essential to QoS networks. We next consider network components needed to support bounded-delay services in a packet-switched QoS network.

1.2 Network Support for Bounded-Delay Services

A QoS network must be *connection-oriented*[1] with a *resource reservation scheme* to ensure the availability of resources such as bandwidth and buffer space for supporting the delay constraints of all traffic. A resource reservation scheme allows the network to quantify the maximum possible traffic before a connection is established and to mathematically verify that all packets will be delivered with the appropriate QoS. In this section, we first discuss network mechanisms needed to support resource reservation, and we then consider issues of network design that impact the degree to which network resources are utilized.

[1]Although the Internet architecture is connectionless, the proposed integrated-services Internet includes the notion of connections. In particular, network resources are reserved for the support of a connection on a fixed path of links and switches using a resource reservation scheme such as RSVP.

1.2.1 Admission Control and Traffic Policing Mechanisms

Two mechanisms are needed in a QoS network that are not used in traditional packet-switched networks to support a resource reservation scheme, namely *admission control* and *traffic policing mechanisms*.[2] We describe both of these mechanisms in the following discussion of the connection-establishment procedure.

We assume that all packets on a single connection traverse the network on a fixed path of switches and links. A client desiring a new connection submits to the network (1) a *traffic characterization* that specifies the maximum traffic on the connection and (2) a *delay bound* that specifies the maximum end-to-end delay to be experienced by any packet on the connection [31]. After a route is determined for the prospective connection, the network employs admission control mechanisms to check whether sufficient bandwidth and buffer space is available along the connection path to support the traffic specified by the traffic characterization at its desired delay guarantee. If sufficient resources are available to ensure that all packets on both the new connection and existing connections will be delivered in accordance with their delay constraints, then the network accepts the new connection and commits to support its delay guarantees throughout its lifetime.

After a connection is established, the network must monitor traffic submitted on the connection with traffic policing mechanisms to ensure that all traffic complies with its negotiated traffic characterization. Traffic policing mechanisms either drop or delay packets that do not conform to the traffic characterization, preventing excessive traffic from entering the network. We note that if the traffic admitted on a connection were to exceed its traffic characterization, delay guarantees for all connections would be compromised.

The mechanisms of a QoS network described above are illustrated in Figure 1.1. In the figure, bold arrows indicate the fixed route of links and switches for the connection established between a sender and receiver. The sender contacts admission control mechanisms

[2] Admission control mechanisms and traffic policing mechanisms are referred to as *Connection Admission Control* (CAC) and *Usage Parameter Control* (UPC), respectively, in the ATM community [1].

before establishing the connection, and all traffic submitted to the network is monitored by traffic policing mechanisms.

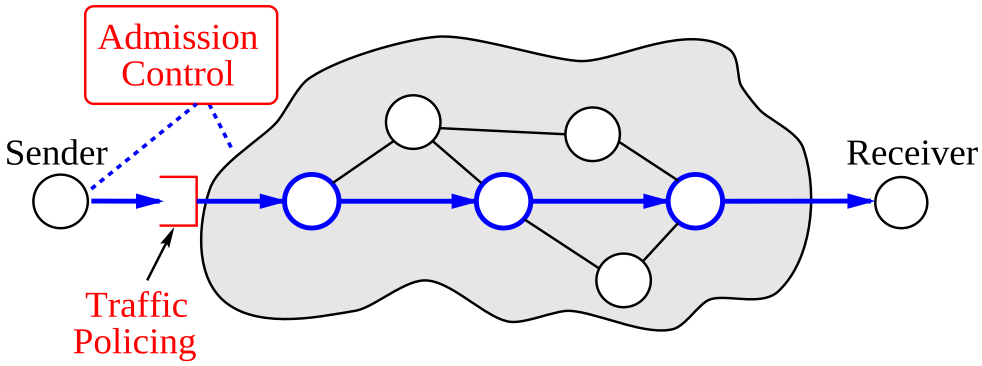


Figure 1.1: QoS network architecture.

Admission control and traffic policing mechanisms restrict the number of connections and the traffic on those connections, thereby limiting the network utilization. Since it is important to maximize the utilization of resources, the network and its resource reservation scheme must be designed such that (1) a large number of connections can be supported with the fixed resources available and (2) resource usage of connections are accurately estimated, resulting in the admission of a large number of connections. The degree to which a network satisfies these two requirements is largely determined by the choice of *traffic characterization methods* and *packet scheduling disciplines*, which we describe in the next section.

1.2.2 Traffic Characterization and Packet Scheduling

Two components that are crucial in the design of a QoS network and significantly impact the overall network utilization are the traffic characterization and packet scheduling discipline. A traffic characterization for a bounded-delay service specifies an upper bound on the traffic of a connection. Since traffic characterizations are used for both admission control and traffic policing mechanisms, the choice of characterization determines a tradeoff between conflicting demands of these two mechanisms: On the one hand, a characterization should be sophisticated enough to provide a precise description of the traffic so that admission control mechanisms do not unnecessarily reject connections due to overestimating their resource requirements. On the other hand, a traffic characterization should conform to a simple traffic model with a small number of parameters so that policing mechanisms can monitor the traffic in real time.

Packets from different connections multiplexed on a single outgoing link of a packet switch are stored in a transmission queue, and the packet scheduler at the switch determines the transmission order of these packets. The set of rules a packet scheduler uses for ordering queued packets is called the packet scheduling discipline (e.g., First-Come-First-Served). Since the queueing delays of packets on a connection are determined by the packet scheduling discipline, *delay bound tests* that verify if packets will be delivered within their delay constraints are developed for the particular scheduling discipline in use at a switch. As we will discuss in Section 2.3 the delay bound test is the most crucial test performed by admission control mechanisms. We also note that the packet scheduler must select packets for transmission at the speed of the transmission link if it is not to become a bottleneck; for this reason, the computational overhead of a scheduling discipline should be limited.

To maximize network utilization in a network with a bounded-delay service, the traffic characterization and packet scheduling discipline should be carefully designed. However, the conventional wisdom is that supporting a bounded-delay service necessitates a peak-rate reservation scheme that will result in low network utilization. As a result, both the Internet and ATM communities have focused on intermediate service classes: predictive services for Internet and ABR services for ATM. Many approaches to traffic characterization do not provide a worst-case bound on the traffic arrivals of a connection and therefore cannot be employed in a bounded-delay service that provides worst-case guarantees [5, 34, 37, 45, 47, 53, 79]. Similarly, many packet schedulers have been proposed that are optimized for criteria other than guaranteeing a bounded delay, e.g., fairness in allocating "spare bandwidth" and isolation of connections [81, 82, 83, 115, 116]. In this dissertation, we consider the design of traffic characterization and packet scheduling methods that are intended explicitly for bounded-delay services [23, 31, 74, 82]. Our goal is to design methods that can be implemented straightforwardly and result in a high network utilization.

1.3 Our Approach to Characterization and Scheduling

The approach of this work is motivated by a study in [103] by Wrege, Knightly, Zhang, and Liebeherr. In this paper, the authors explore network utilization limits of a deterministic service by combining the tightest traffic characterizations and the best-possible packet scheduling discipline with its necessary and sufficient delay bound tests. In particular, they used the so-called *empirical envelope* of a traffic source for traffic characterization and the *Earliest-Deadline-First* (EDF) packet scheduling discipline [31] with delay bound tests developed in [74]. They showed that these components result in maximal network utilization, and they demonstrated empirically that a reasonable utilization (i.e., around 40%) can be obtained when supporting VBR video traffic sources with delay bounds on the order of 10-50 milliseconds.

Although the empirical envelope and EDF scheduler can be used to achieve the highest-possible network utilization in a bounded-delay service, both of these components have drawbacks which make them impractical for use in a QoS network. Many drawbacks of the empirical envelope, which we discuss in detail in Chapter 4, are due to its large number of necessary parameters. The task of determining these parameters requires significant computation, and so the production of the empirical envelope is expensive and cannot be performed in real time. Further, the empirical envelope cannot be policed with simple traffic policing mechanisms since it does not conform to a simple traffic model. Finally, since the complexity of delay bound tests is a function of the traffic characterization, admission control mechanisms may not be able to verify resource availability in a small amount of time.

With regard to scheduling, EDF requires the sorting of packets into a single transmission queue, a task which is prohibitively expensive in high-speed networks. For these reasons, neither the empirical envelope nor EDF can be used in a QoS network.

Our approach is to produce practical traffic characterization methods and packet scheduling disciplines that approximate the achievable utilization of the empirical envelope and EDF scheduler, respectively. For traffic characterization, we choose a powerful traffic model that can be easily enforced by policing mechanisms and select parameters for this model such that the resulting characterization can admit almost as many connections as the empirical envelope itself. For packet scheduling, we design a novel scheduler that approximates EDF without requiring the complex sorting operation. This dissertation makes contributions in both of these areas, and the thesis statement of this research is as follows:

> *By carefully designing traffic characterization and packet scheduling methods, packet-switched networks can provide worst-case QoS guarantees while maintaining high network utilization.*

1.4 Structure of the Dissertation

In this dissertation, we present traffic characterization methods and packet scheduling disciplines that are practical for implementation yet yield a network utilization that approximates the optimal. We first discuss the network framework and study tradeoffs between overhead and achievable utilization for previous methods. We then present a characterization method that is designed to approximate an optimal characterization while requiring only a small number of computations. We finally present a novel scheduling discipline that provides a tradeoff between high utilization and low overhead costs, and we demonstrate that our scheduler can achieve both to a high degree.

In Chapter 2 we review the framework of a bounded-delay service and describe its three key components: traffic characterization, packet scheduling discipline, and delay bound tests. This chapter states our assumptions on the network architecture and presents properties needed for our theoretical contributions.

In Chapter 3 we review related work, focusing first on deterministic traffic characterizations and their respective policing mechanisms and then on packet scheduling disciplines that can provide deterministic delay guarantees. In the context of bounded-delay services, we discuss tradeoffs between implementation overhead and achievable network utilization for different methods from the literature.

Chapter 4 presents a novel method for traffic characterization of MPEG-compressed video. In particular, the method tries to produce a characterization that closely approximates the optimal traffic characterization for a traffic source, its so-called *empirical envelope*. The empirical envelope has two drawbacks that make it impractical for use in a deterministic service: first, it requires a large number of parameters that are computationally expensive to compute, and second, it cannot be monitored with simple policing mechanisms. We address each of these problems in turn. We first show how to reduce the number of parameters of the empirical envelope needed for traffic characterization using an extrapolation technique. We next use our reduction as the basis of an algorithm that selects parameters for an easily-enforced traffic model to closely approximate the empirical envelope itself. We also devise a scheme in which connections can dynamically renegotiation their traffic rates, and we apply our characterization method to this renegotiation scheme.

In Chapter 5 we present a novel packet scheduling discipline called Rotating-Priority-Queues^{+} (RPQ^{+}). The RPQ^{+} scheduler is designed to approximate the optimal EDF packet scheduler. However, unlike the EDF scheduler which requires sorting operations that make it impractical for use in high-speed networks, RPQ^{+} can be implemented with simple operations and allows for a smooth tradeoff between overhead costs and achievable utilization. We analyze the RPQ^{+} scheduler and derive its necessary and sufficient schedulability conditions that can be used in admission control mechanisms. We demonstrate that

(1) RPQ^+ can always achieve a utilization superior to that of a priority scheduler and (2) RPQ^+ can approximate EDF with arbitrary precision.

We present our conclusions and summarize the contributions of this dissertation in Chapter 6. We also outline future research directions.

2

Framework of a Bounded-Delay Service

As motivated in Chapter 1, a network with a bounded-delay service requires a resource reservation scheme to allocate network resources for individual connections. Two mechanisms crucial to the design of a resource reservation scheme are the admission control mechanisms that limit the number of admitted connections and the traffic policing mechanisms that limit the traffic on individual connections. These mechanisms should be designed such that the network admits a large number of connections, resulting in a high network utilization. In this chapter, we describe the three components central to admission control and traffic policing mechanisms that impact the achievable network utilization: *traffic characterization*, *packet scheduling disciplines*, and *delay bound tests*.

2.1 Traffic Characterization

To quantify the traffic on connections, a QoS network uses a traffic characterization for each connection using the bounded-delay service. A traffic characterization appropriate for use in a bounded-delay service must satisfy several requirements. First, the characterization must provide a *worst-case* description of the source that determines an upper bound on a source's packet arrivals. Second, the characterization must conform to a parameterized *traffic model* so that a source can efficiently specify its traffic characterization to the network with few parameters. Third, the traffic model must be *policeable*, that is, it must conform to traffic policing mechanisms which are easily implemented so that the network can enforce a source's traffic characterization. Finally, the traffic characterization should be sophisticated enough to describe the traffic *accurately* so that the admission control mechanisms do not overestimate the resources required by the connection.

Since a deterministic service provides worst-case guarantees, a traffic characterization must specify the worst-case traffic of a connection. We let A denote the actual traffic on a connection, where $A[\tau, \tau + t]$ denotes the traffic arrivals in time interval $[\tau, \tau + t]$. Then, a worst-case characterization of the traffic A is given by a *traffic constraint function* A^* which provides an upper bound on A. A traffic constraint function A^* should satisfy two important properties, namely *time-invariance* and *subadditivity* [23, 74]. A function A^* provides a time-invariant bound for A if for all times $\tau \geq 0$ and $t \geq 0$ the following holds [23]:

$$A[\tau, \tau + t] \leq A^*(t) \tag{2.1}$$

Since a time-invariant traffic constraint function A^* bounds the maximum traffic over any time interval of length t, the delay-bound tests can be made independent of the starting time of a connection. A traffic constraint function A^* is subadditive if it satisfies the following inequality:

$$A^*(t_1) + A^*(t_2) \geq A^*(t_1 + t_2) \qquad \forall t_1, t_2 \geq 0 \tag{2.2}$$

A subadditive traffic constraint function allows the arrivals on a connection to attain the bound given by A^*. In other words, it is feasible that $A[\tau, \tau + t] = A^*(t)$ for any $t \geq 0$. Even though traffic constraint functions that are time-invariant but not subadditive have been proposed, e.g., [58], we point out that any such traffic constraint function A_1^* can be replaced by a subadditive function A_2^* such that $A_2^*(t) \leq A_1^*(t)$ for all $t \geq 0$. Finally, we wish to add that admission control mechanisms for QoS networks generally assume that traffic constraint functions are both time-invariant and subadditive [23, 74, 82]. In this dissertation, we call a traffic constraint function A^* for A *viable* for a deterministic service if it satisfies both equations (2.1) and (2.2).

Practical traffic characterizations are obtained from a parameterized *traffic model* which in turn expresses the maximum traffic admitted by some traffic policing mechanism. For example, consider the (σ, ρ) traffic model [23] which describes the worst-case traffic admitted by a leaky bucket mechanism with a burstiness parameter σ and a rate parameter ρ. We denote the traffic constraint function that provides a bound on the maximum traffic conforming to the (σ, ρ)-model by B^*, where B^* is given by the following linear constraint [23]:

$$B^*(t) = \sigma + \rho t \qquad \text{for all } t \geq 0 \tag{2.3}$$

Figure 2.1 depicts traffic A that conforms to the (σ, ρ)-model and its corresponding traffic constraint function B^*. Note that B^* is an upper bound on A. Use of such a traffic model is essential because (a) it allows a simple interface between the network and its clients since the traffic characterizations can be specified with a small number of parameters and (b) it guarantees that the traffic characterization can be easily enforced by policing mechanisms. We defer discussion of traffic models and traffic policing mechanisms that are currently in use to Chapter 3.

Although the use of traffic models is essential to practical traffic characterization, many traffic models are not sophisticated enough to characterize a variable bit rate (VBR) traffic source. In particular, video traffic compressed with the MPEG compression algorithm has complex and seemingly irregular timely correlations (see Figure 2.2) that are difficult to characterize accurately with a function of few parameters. The shape of a traffic constraint function depends on the selection of the traffic model. While in general, a model with more parameters can achieve a more accurate or tight traffic constraint function, the additional parameterization causes an increase in the complexity of policing the traffic model. Thus, the selection of an appropriate traffic model for a deterministic service must find a compromise between the high complexity preferred by the admission control mechanisms and the simplicity required for the implementation of traffic policing mechanisms.

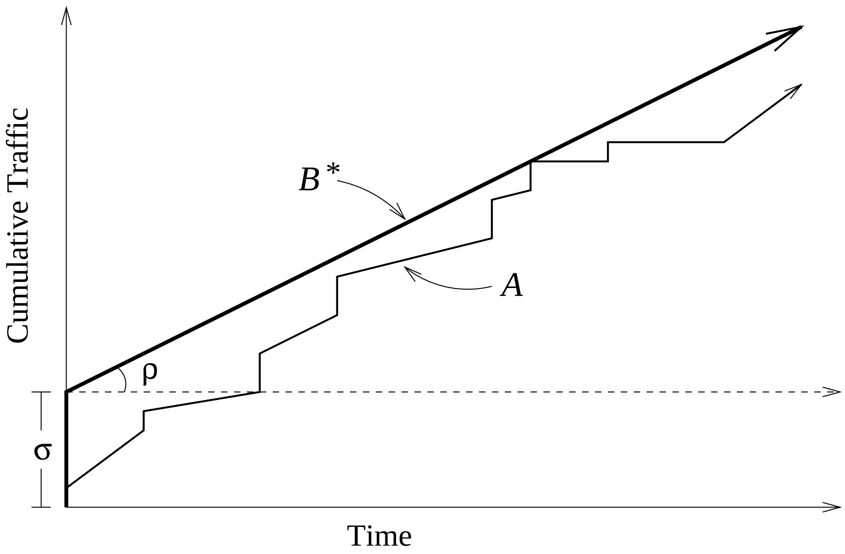


Figure 2.1: Traffic A and traffic constraint function B^*.

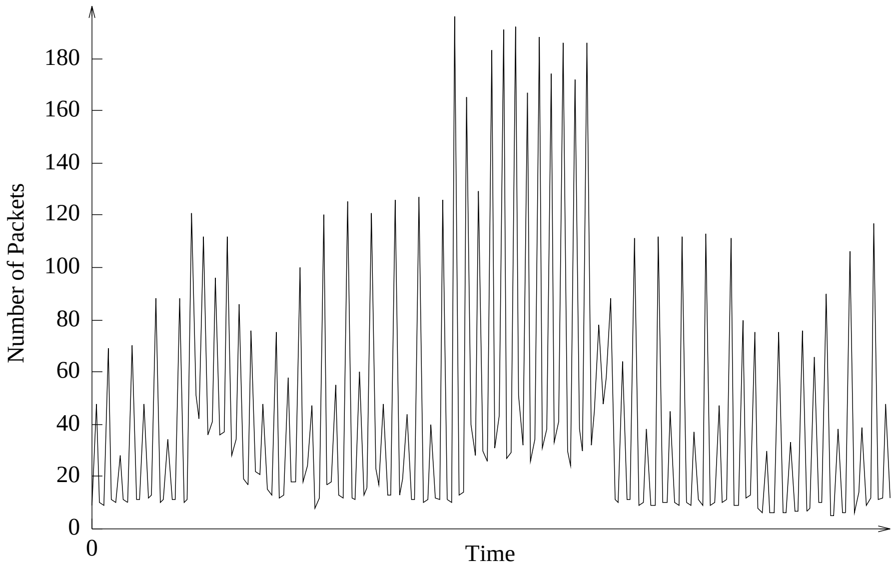


Figure 2.2: Traffic of an MPEG video stream.

2.2 Packet Scheduling Discipline

The packet scheduler, is central in controlling the end-to-end delay of packets in a QoS network. Multiplexed streams of packets arrive to a switch on incoming links where the packets are demultiplexed and switched to outgoing links based on their connections. Packets waiting to be transmitted on an outgoing link are stored in a *transmission queue* for the link, and the packet scheduler at the switch determines the transmission order of queued packets according to the packet scheduling discipline. Figure 2.3 illustrates the paths of packets through a single network switch. In the figure, packets are assumed to flow from left to right, and packets from two connections A and B are depicted awaiting transmission on the same outgoing link; the packet scheduling discipline at the switch determines the packet to be transmitted first.

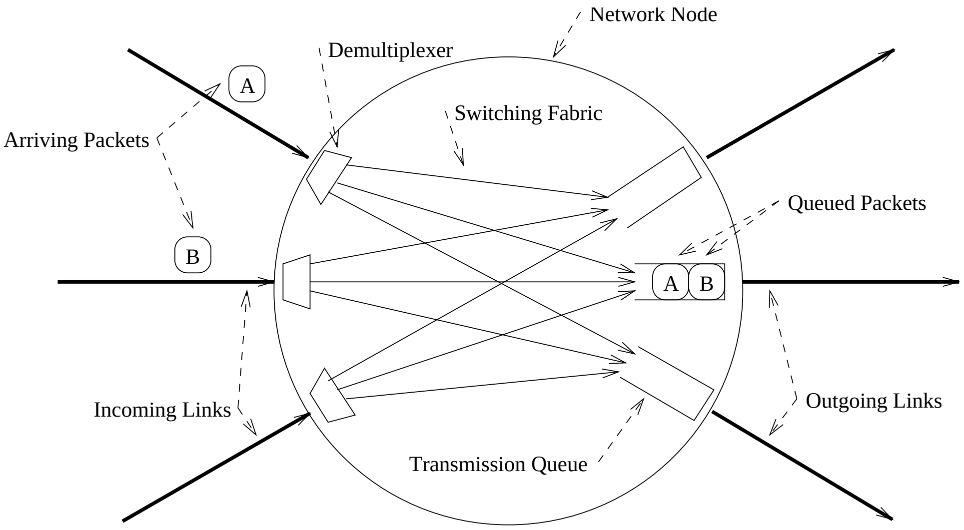


Figure 2.3: Path of packets through a network switch.

Packets from different connections may interact with one other by contending for the same transmission link, and the packet scheduling discipline determines the *queueing delays* of packets, that is, the time a packet spends in the transmission queue. A packet scheduling discipline is classified as either *work-conserving* or *non-work-conserving*. A work-conserving packet scheduler is never idle if packets are waiting in the transmission queue, while a non-work-conserving scheduler may be idle even if queued packets are available. A scheduling discipline may also be classified as *preemptive* or *non-preemptive*. A preemptive scheduling discipline may suspend the transmission of one packet in order to transmit another, while a non-preemptive discipline does not interrupt the transmission of packets. In this dissertation, we consider packet schedulers that are both work-conserving and non-preemptive. In this case, the only two instants when a scheduler selects a packet for transmission are (a) after the completion of a packet transmission if the transmission queue is non-empty and (b) after a packet arrival at an empty scheduler.

In the presence of admission control and traffic policing mechanisms, a large number of packet scheduling disciplines can provide bounds on delays [30]; however, most schedulers will result in an inefficient use of network resources. The performance of a packet scheduler in providing bounded delay services can be determined by the degree to which it satisfies the following requirements [74]:

- *Efficiency:* A high utilization of network resources such as link bandwidth can only be achieved if the packet schedulers can support bounded delays for a large number of connections.

- *Flexibility:* A packet scheduler must be sufficiently flexible to satisfy a diverse set of delay requirements. For example, a FIFO scheduler can support only one delay bound for all connections and thus has insufficient flexibility.

- *Complexity:* Since multiplexing of packets must be performed at the speed of the transmission link, the complexity of the packet scheduling discipline must be kept minimal. If the operations at the packet scheduler consume more time than the actual transmission of a packet, transmission links will be left idle most of the time.

- *Analyzability:* The admission control mechanisms which determine whether a new connection may result in delay bound violations of requested or existing connections require analytical *schedulability conditions* for the packet schedulers, that is, expressions which determine if the maximum delay of any packet may exceed its delay bound. If exact schedulability conditions are not available, the admission control mechanisms will unnecessarily limit the number of connections in the network and reduce the efficiency of the packet scheduler. We will discuss the use of schedulability conditions in delay bound tests in Section 2.3.

2.3 Delay Bound Tests

A connection i that traverses a set of n network switches $\{1, 2, \ldots, n\}$ has an *end-to-end delay bound* D_i which denotes the maximum tolerable network latency for any packet on connection i. A packet submitted to the network on connection i at time t is assigned a *deadline* of $t + D_i$. A *deadline violation* occurs if any packet is not delivered to the destination before its deadline. The delay bound test is a set of conditions that, when satisfied, guarantees deadline violations will not occur.

We decompose the problem of controlling the end-to-end latency across multiple switches by considering the delay at each individual switch. The delay bound D_i is thus divided into a set of *local delay bounds* $\{d_{i,j}\}_{1 \le j \le n}$, where $d_{i,j}$ specifies the maximum delay for connection i across the jth switch and $\sum_{j=1}^{n} d_{i,j} = D_i$. There are several approaches to partitioning the end-to-end delay bound D_i into local delay bounds [8, 14, 94]. When we consider connections at a single switch, we drop the subscript j and denote that delay bound for connection i by d_i. Similar to the end-to-end formulation, a packet arriving to a switch at time t on connection i is assigned a local deadline of $t + d_i$ and has a deadline violation if it is not fully transmitted by the packet scheduler before its deadline. The end-to-end delay bound test is decomposed into a series of local delay bound tests at each switch along the path of the connection. Since propagation and processing delays are largely fixed due to physical constraints, we assume for clarity of presentation that these delays are zero, and so d_i is a bound on the sum of the queueing delay and transmission time.

For a given packet switch, we say that a set $\mathcal{C}$ of connections with traffic constraint functions and delay bounds $\{A_i^*, d_i\}_{i \in \mathcal{C}}$ is *schedulable* if a deadline violation cannot occur for any connection i that conforms its traffic to A_i^* as shown in equation (2.1). The conditions which determine if a set of connections is schedulable, called *schedulability conditions*, constitute the delay bound test in bounded-delay services. A delay bound test depends heavily upon the choice of packet scheduler and traffic model since tests are developed specifically for each packet scheduler and use traffic constraint functions as arguments. Thus, the properties of the packet scheduler and the accuracy of the traffic model are directly reflected in the delay bound test.

The case of connections that traverse multiple switches is nontrivial since the traffic may become distorted at downstream switches. In particular, traffic entering the network on a connection i that conforms to A_i^* may not conform to A_i^* at downstream switches after interacting with traffic from other connections. However, multi-hop routes can be addressed by either quantifying the distortion of the worst-case traffic arrivals A_i^* at different switches [24] or controlling the distortion of the arrivals by reshaping the traffic to conform to A_i^* at each switch with so-called traffic shaping mechanisms [109]. In the remainder of this dissertation we restrict our attention to the delay at a single network switch.

3

Previous Work

Although there is a great deal of related work on both traffic characterization and packet scheduling, much of this work cannot be directly applied to the support of multimedia traffic over a network with a bounded-delay service. Regarding video traffic characterization, many approaches characterize traffic sources using sophisticated stochastic models such as Markov-modulated [47], autoregressive [48, 50, 59], self-similar [5, 37], TES [50, 79], and S-BIND [60, 113]. These approaches do not do not provide a worst-case bound on traffic arrivals and therefore cannot be used as a traffic characterization as described in Chapter 2.1. Additionally, it is difficult to design simple policing mechanisms to enforce statistical properties of a traffic source in real time [10]. With regard to packet scheduling, research from the real-time community (e.g., [67, 68, 76, 97]) uses assumptions that are inappropriate for QoS networks, i.e., connections are assumed to have periodic packet arrivals where all packets on a connection have identical service times and a delay bound equal to the period. The system is also assumed to be preemptive. However, in a communications network, traffic is bursty with delay constraints independent of the traffic rate, and the transmission of a packet cannot be preempted.

In this chapter we review approaches to traffic characterization and packet scheduling suitable for use in networks with a bounded-delay service. In Section 3.1 we review a number of deterministic traffic models [23, 24, 31, 41, 55, 103]. We consider both the traffic constraint function that specifies the maximum traffic conforming to each model as well as available traffic policing mechanisms for enforcement. We discuss the advantages and disadvantages of each traffic characterization method. Section 3.2, where we describe the operations and implementations of a number of packet schedulers and review their available delay bound tests [31, 32, 33, 40, 52, 63, 74, 82, 102, 107, 110, 111, 112]. We describe the properties of each packet scheduling discipline, emphasizing tradeoffs between achievable network utilization and implementation overhead costs.

3.1 Traffic Characterization

A traffic characterization should conform to a parameterized traffic model that can be enforced by some traffic policing mechanism. To be used in the delay bound test of a bounded-delay service, a characterization must specify the maximum traffic on a connection. In this section we review six such traffic models which have been considered for use in bounded-delay services: the peak-rate model [3], the (σ, ρ)-model [23], the $(\vec{\sigma}, \vec{\rho})$-model [24, 103], the (r, T)-model [41], the $(X_{min}, X_{ave}, I, s^{max})$-model [29, 31], and the D-BIND model [55, 58]. We formulate the traffic constraint function A^* for each traffic model and also discuss traffic policing mechanisms used for enforcement [26, 86, 87]. We then review studies that evaluate and compare the tradeoffs involved with each traffic model.

3.1.1 Peak-rate model

The peak-rate model is the simplest and most widely used of all traffic models [6]. For this model, two parameters are used to describe traffic on a connection: a minimum interarrival time X_{min} and the maximum transmission time s^{max} of any packet. The maximum traffic for a connection that conforms to the peak-rate model is given by the following traffic constraint function:

$$A^*(t) = \left(\left\lfloor \frac{t}{X_{min}} \right\rfloor + 1 \right) s^{max} \qquad \text{for all } t \geq 0 \tag{3.1}$$

Often a parameter ρ is used to express the maximum arrival rate of a connection, where $\rho \equiv s^{max}/X_{min}$. The peak rate model can be easily enforced by a *packet spacing mechanism* that ensures a minimum interarrival time between consecutive packets [1, 12, 101]. Note that the peak-rate model specifies CBR traffic and will overestimate resource requirements of a VBR source.

3.1.2 (r, T)-model

The (r, T)-model describes its traffic with a rate parameter r and a framing interval T [41]. Time is partitioned into frames of length T, and the maximum traffic on a connection during any frame is limited to rT bits. Thus, the (r, T)-model enforces an average rate r while allowing for moderate bursts. Note that r specifies the maximum average transmission rate for a compliant connection.

Traffic conforming to the (r, T)-model can be policed with the *jumping window* policing mechanism which is implemented as follows [87]. At the beginning of each frame, a credit counter is set to rT. This counter is decremented by 1 for each bit that enters the network, and packets can only enter the network if sufficient credit is available, i.e., the counter must always be nonnegative. Every T time units, the credit variable is reset to rT. If a packet arrives whose admission would result in a negative counter value, it is not allowed into the network and will be either discarded or queued until the next frame. This mechanism is illustrated in Figure 3.1, where we plot the value of the counter over a period of 5 frame times. In the figure, the credit variable is reduced whenever a packet arrives to the scheduler. Note that the credit variable is not depleted during the first interval, but it is shown to fully drain in the second interval.

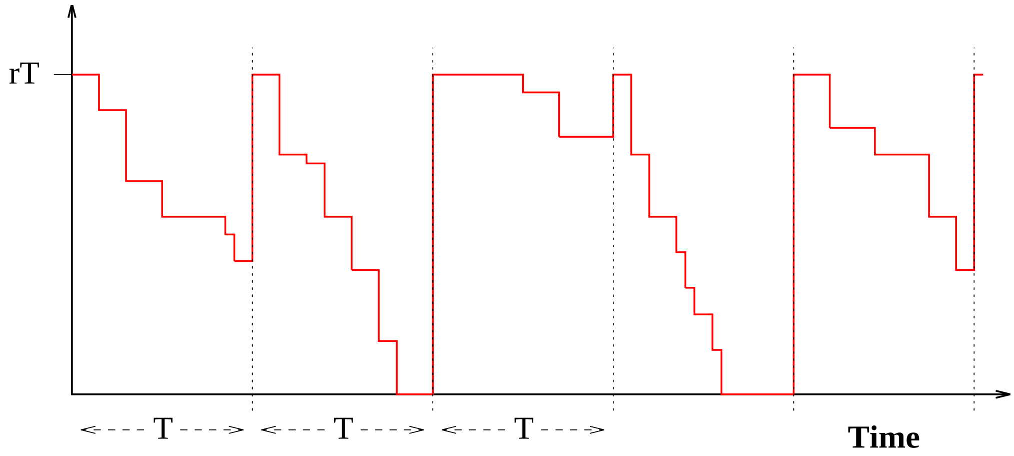


Figure 3.1: Illustration of the jumping window mechanism for the (r, T)-model.

The traffic constraint function for the (r, T)-model is given by the following:

$$A^*(t) = \left(\left\lceil \frac{t}{T} \right\rceil + 1 \right) rT \qquad \text{for all } t \geq 0 \tag{3.2}$$

Observe that the maximum burst that may enter the network is $2rT$. Such a burst occurs if a burst of rT occurs immediately prior to the beginning of a frame and another burst of size rT occurs at the beginning of the frame. The traffic constraint function A^* is illustrated in Figure 3.2.

Note that the parameters r and T must be selected such that the product rT bounds the arrivals on a connection over any interval of length T, that is, $rT \geq A[t, t+T]$ for any $t \geq 0$. Thus, the rate r, which will vary between the peak and average bit rate of the connection, is dependent on the choice of frame length T. Small values of T result in a rate r close to the peak rate of the connection, while larger values of T result in smaller rates r, approaching the average rate of the connection in the limit. Observe also in Figure 3.2 that the worst-case burst of size $2rT$ is twice as large as the actual burst rT of traffic on the connection.

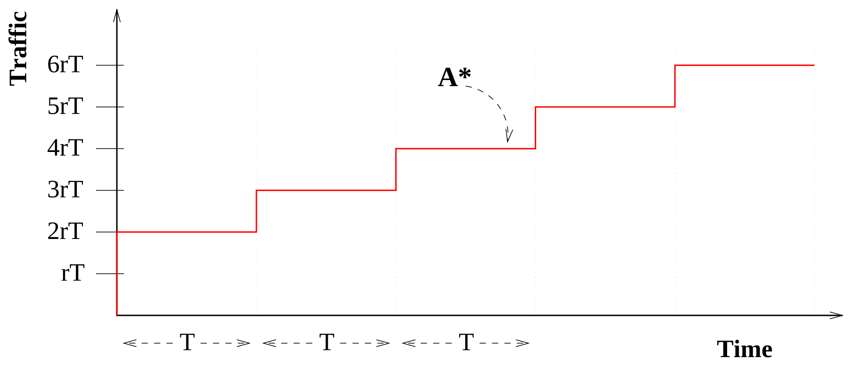


Figure 3.2: Worst-case bound A^* for the (r, T)-model.

3.1.3 (σ, ρ)-model

The (σ, ρ)-model describes its traffic with a burst parameter σ and a rate parameter ρ [23]. The traffic on a connection over any time interval of length t is limited to $\sigma + \rho t$. The (σ, ρ)-model is the traffic model for the well-known *leaky bucket* policing mechanism[1] which is implemented as follows [100]. A credit counter is initialized to σ, and traffic may only enter the network if the counter is nonzero. The credit counter is decremented for each bit that enters the network, and the counter is continuously incremented at rate ρ when its value is less than σ. In Figure 3.3 we illustrate the credit counter of a leaky bucket mechanism. Note in the figure that the credit counter is reduced by the packet transmission times of new packet arrivals and that it is always increased at rate ρ when its value is less than σ. This model enforces a rate ρ while allowing some burstiness up to σ. Efficient implementations of the leaky bucket mechanism are discussed in [1, 100].

[1] In the ATM community, a leaky bucket mechanism is referred to as the Generic Cell Rate Algorithm (GCRA) [1].

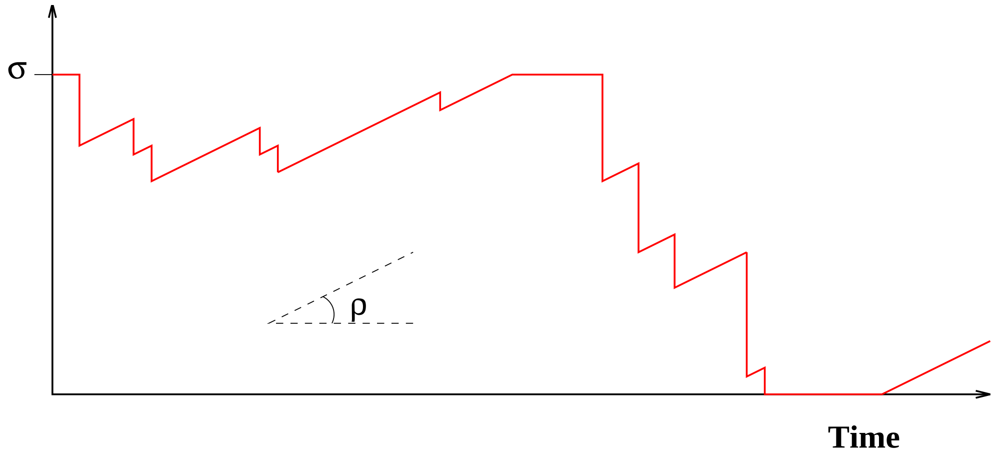


Figure 3.3: Illustration of the leaky bucket mechanism for the (σ, ρ)-model.

We use B^* to denote the traffic constraint function for the (σ, ρ)-model, where B^* is obtained straightforwardly as follows:

$$B^*(t) = \sigma + \rho t \qquad \text{for all } t \geq 0 \tag{3.3}$$

In Figure 3.4 we illustrate the traffic constraint function B^*.

Note that the (σ, ρ) model is more flexible than the (r, T)-model since the burstiness parameter σ is independent of the rate ρ. The maximum burst admitted by the (r, T)-model is proportional to the rate and is given by $2rT$.

3.1.4 $(\vec{\sigma}, \vec{\rho})$-model

A generalization of the (σ, ρ)-model is the $(\vec{\sigma}, \vec{\rho})$ traffic model [24, 103] which corresponds to a traffic policing mechanism where multiple leaky buckets are connected in series. For a connection that conforms to the $(\vec{\sigma}, \vec{\rho})$-model with a set of m pairs $\{(\sigma_i, \rho_i)\}_{1 \leq i \leq m}$, the amount of traffic admitted to the network is limited by each of the (σ_i, ρ_i) pairs. The resulting traffic constraint function, denoted as B^*_m, is a function consisting of m piecewise-linear

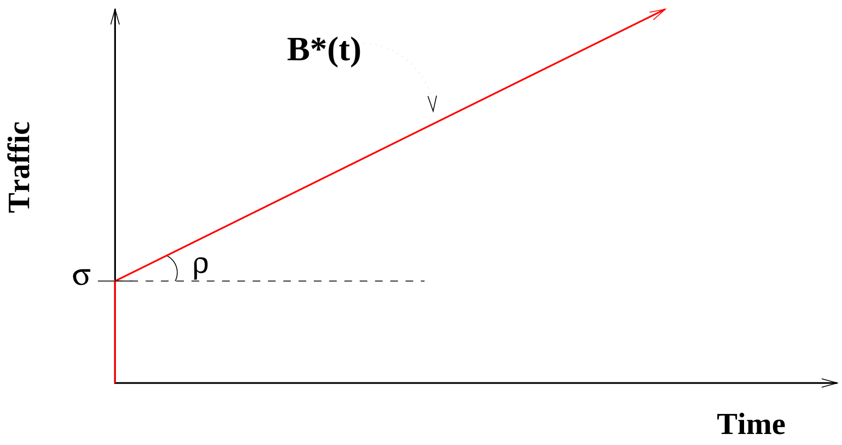


Figure 3.4: Worst-case bound B^* for the (σ, ρ)-model.

segments [24, 103]:

$$B_m^*(t) = \min_{1 \le i \le m} \{\sigma_i + \rho_i t\} \qquad \text{for all } t \ge 0 \tag{3.4}$$

Note that B^* in equation (3.3) is identical to B_1^* in equation (3.4). We observe from equation (3.4) that B_m^* is a concave function, that is, $B_m^*[\tau_1, \tau_1 + t] \ge B_m^*[\tau_2, \tau_2 + t]$ for all $\tau_1 \le \tau_2$. In Figure 3.5 we illustrate the traffic constraint function B_m^* for three (σ, ρ) pairs.

The $(\vec{\sigma}, \vec{\rho})$-model has been employed in real systems [2, 95]. For example, the ATM Forum specifies that its connections are to use two (σ_i, ρ_i) pairs, where the first of the two pairs is set such that $\sigma_1 = 0$ and ρ_1 is equal to the peak traffic rate [2].

3.1.5 $(X_{min}, X_{ave}, I, s^{max})$-model

In the $(X_{min}, X_{ave}, I, s^{max})$-model [29, 31], X_{min} is the minimum packet interarrival time, X_{ave} is the maximum average packet interarrival time over any time interval of length I, and s^{max} is the maximum packet transmission time. This traffic model thus limits the peak rate of a connection while ensuring that the traffic admitted during any interval of length I

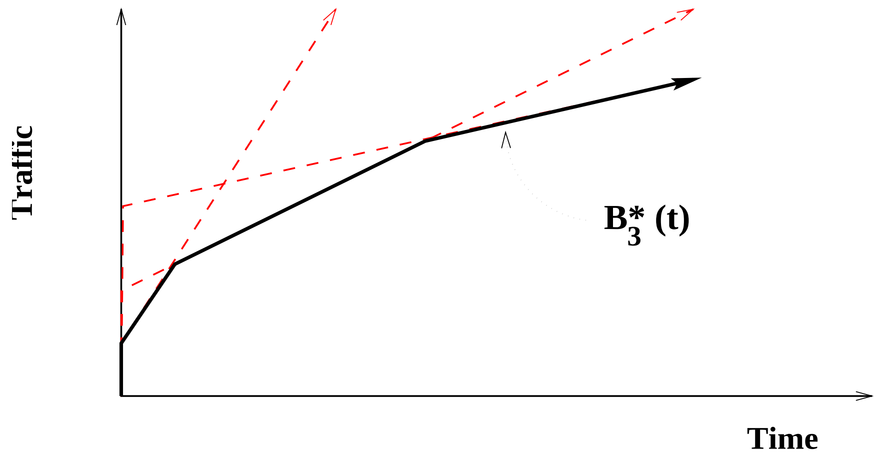


Figure 3.5: Worst-case bound B^*_m for the $(\vec{\sigma}, \vec{\rho})$-model. In the figure, the number of linear segments is set to $m = 3$.

is at most $\frac{I \cdot s^{max}}{X_{ave}}$. The traffic constraint function for the $(X_{min}, X_{ave}, I, s^{max})$-model, illustrated in Figure 3.6, is given as follows:

$$A^*(t) = \lfloor \frac{t}{I} \rfloor \cdot \frac{I \cdot s^{max}}{X_{ave}} + \min \left\{ \left\lceil \left(\frac{t}{I} - \lfloor \frac{t}{I} \rfloor \right) \cdot \frac{I}{X_{min}} \right\rceil , \frac{I}{X_{ave}} \right\} \cdot s^{max} \qquad \text{for all } t \geq 0 \quad (3.5)$$

In addition to a cell spacing mechanism that enforces the minimum interarrival time, the $(X_{min}, X_{ave}, I, s^{max})$-model requires a *moving window* mechanism for its enforcement. The moving window is similar to the jumping window described in Section 3.1.2 in that packet arrivals are limited over intervals of length I. However, here time is not divided into frames, and so each packet arrival must be remembered for exactly I time units. A counter is initialized to $\frac{I \cdot s^{max}}{X_{ave}}$, and, for each packet with transmission time s that arrives, the counter is decreased by s. Exactly I units after the arrival instant of the packet, the counter is increased by s, and so the moving window mechanism requires packet arrivals to be stored for I time units. A packet is not admitted if its admission would result in a negative counter value. We illustrate the maintenance of the counter in Figure 3.7. In

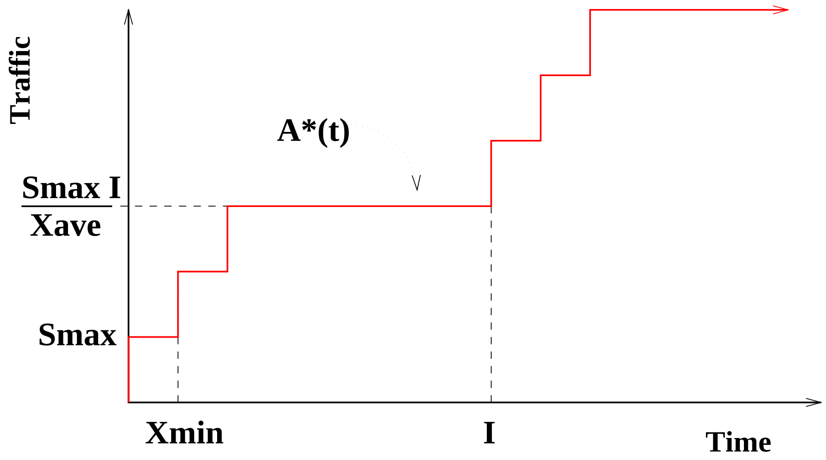


Figure 3.6: Worst-case bound A^* for the $(X_{min}, X_{ave}, I, s^{max})$-model.

the figure, the value of the counter is decreased for each packet arrival, and we mark the intervals of length I for which the first two packet arrivals must be remembered.

3.1.6 D-BIND model

The D-BIND traffic model is a general traffic model that uses a number of rate-interval pairs $\{(R_i, I_i) | i = 1, \ldots n\}$ [55, 58]. The maximum rate over any interval of length I_i is restricted to R_i for all pairs i. The traffic constraint function A^* for the D-BIND model is given as follows [55]:

$$A^*(t) = \frac{R_i I_i - R_{i-1} I_{i-1}}{I_i - I_{i-1}} + R_i I_i \qquad \text{for all } I_{i-1} \leq t \leq I_i \tag{3.6}$$

The D-BIND model thus defines an n segment piecewise-linear traffic constraint function. Note that the $(\vec{\sigma}, \vec{\rho})$-model can be viewed as a special case of the D-BIND model since the $(\vec{\sigma}, \vec{\rho})$-model defines an n segment *concave* piecewise-linear traffic constraint function. We also point out that the D-BIND traffic model is distinct from the previous models in

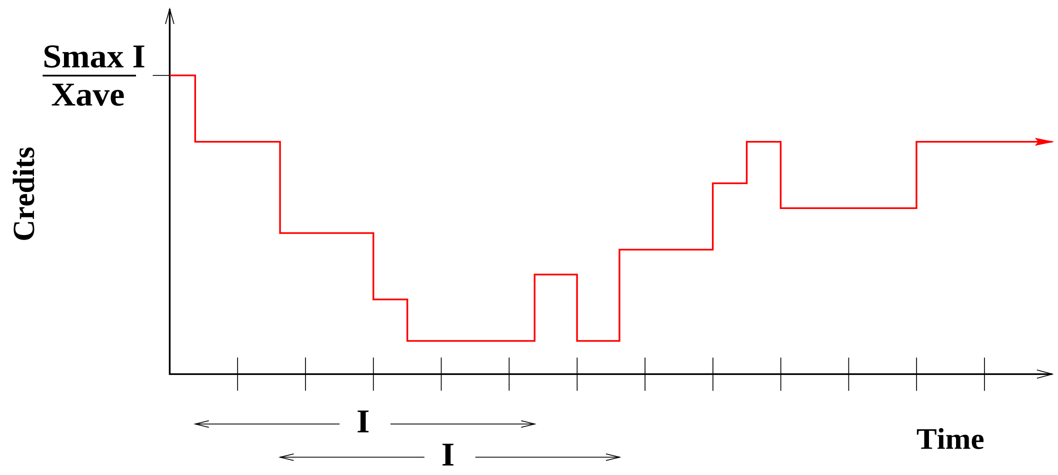


Figure 3.7: Illustration of the moving window mechanism for the $(X_{min}, X_{ave}, I, s^{max})$-model. When the counter is decreased by some Δ at time t, the counter is later increased by the same amount at time $t + I$.

that some instances may not satisfy the subadditivity property discussed in Chapter 2.1. Traffic policing for the D-BIND model requires n moving window mechanisms, one for each rate-interval pair.

3.1.7 Discussion of Tradeoffs

Although the previous traffic models have all been considered for use in bounded-delay services, the choice of a particular traffic model must consider both of the following requirements:

- The model should accurately describe the variable-bit-rate traffic arrivals of a compressed video sequence.

- The policing mechanisms for enforcing the traffic model should have simple implementations.

We next review the results of several studies that have evaluated the various traffic models with respect to these two requirements [26, 86, 87, 88, 103]. We note that these studies do not focus on the six traffic models we just discussed but rather consider the traffic policing mechanisms upon which these traffic models are based: jumping window, moving window, and leaky bucket. For reference, we summarize the relationship between the traffic models and their policing mechanisms in Table 3.1.

Traffic Model	**Policing Mechanisms**	**Traffic Constraint Function**
Peak-rate	Packet spacer	$A^*(t) = (\lfloor \frac{t}{X_{min}} \rfloor + 1)s^{max}$
(r, T)	Jumping window	$A^*(t) = (\lceil \frac{t}{T} \rceil + 1)rT$
(σ, ρ)	Leaky bucket	$B^*(t) = \sigma + \rho t$
$(\vec{\sigma}, \vec{\rho})$	Multiple leaky buckets	$B^*_m(t) = \min_{1 \leq i \leq m}\{\sigma_i + \rho_i t\}$
$(X_{min}, X_{ave}, I, s^{max})$	Moving window and spacer	$A^*(t) = \lfloor \frac{t}{I} \rfloor \cdot \frac{I \cdot s^{max}}{X_{ave}} +$ $+ \min\{\lceil (\frac{t}{I} - \lfloor \frac{t}{I} \rfloor) \cdot \frac{I}{X_{min}} \rceil, \frac{I}{X_{ave}}\} \cdot s^{max}$
D-BIND	Multiple moving windows	$A^*(t) = \frac{R_i I_i - R_{i-1} I_{i-1}}{I_i - I_{i-1}} + R_i I_i$

Table 3.1: Summary of traffic models.

A study by Reibman and Berger [88] and another set by Rathgeb [86, 87] evaluate the accuracy with which the mechanisms can characterize VBR video. Rathgeb shows how the parameters of each traffic policing mechanism can be expressed in terms of parameters of the other mechanisms, enabling a direct comparison of the various mechanisms. The examples presented indicate that the leaky bucket is superior to both of the windowing mechanisms for describing VBR video since neither the jumping window nor the moving window are capable of capturing the short-term burstiness. In [88], which compares only the leaky bucket and moving window, Reibman also concludes that the leaky bucket is the more accurate mechanism for VBR video. However, both studies also note that the use of a single leaky bucket which employs only one rate parameter cannot achieve acceptable accuracies. However, it was shown in [103] that the $(\vec{\sigma}, \vec{\rho})$-model which employs multiple leaky bucket mechanisms can accurately characterize VBR video.

Both Dittmann [26] and Rathgeb [86] consider the implementation complexity of the traffic policing mechanisms. These studies showed that the moving window mechanism is significantly more difficult to implement than either the jumping window or leaky bucket mechanisms. We also note that efficient implementations of the moving window mechanism must be optimized for a single value of I, and so the mechanism is inflexible [86]. Conversely, since the jumping window and leaky bucket can be implemented with similar overhead costs and require less state information (i.e., the entire state is determined by a counter value and a timer), these mechanisms allow for flexible dimensioning.

For these reasons, the networking community has focused primarily on characterizations that can be policed by leaky bucket mechanisms, that is, the (σ, ρ) and $(\vec{\sigma}, \vec{\rho})$ traffic models. We note that both the Internet community [13] and the ATM community [1, 2] specify traffic characterizations that comply with the $(\vec{\sigma}, \vec{\rho})$-model. In our study presented in Chapter 4, all traffic characterizations considered conform to the $(\vec{\sigma}, \vec{\rho})$ traffic model.

3.2 Packet Scheduling and Delay Bound Tests

Packet scheduling disciplines that have been considered for bounded-delay services can be broadly categorized into two classes: *delay-based* disciplines that provide maximum delay guarantees to connections, and *rate-based* disciplines that provide minimum throughput guarantees.[2] Delay-based schedulers are distinguished in that they employ a delay bound d_i when scheduling packets from connection i. For example, a *Static-Priority* scheduler assigns a delay bound d_i to each connection and prioritizes queued packets according to their delay constraints. Admission control mechanisms for delay-based schedulers calculate delay guarantees for a connection based on the traffic from all connections and properties of the packet scheduler. Conversely, rate-based schedulers allocate a fraction of the available bandwidth to each connection and calculate delay guarantees based on their traffic characterizations. For example, a Round-Robin scheduler uses a cyclic service strategy where time is divided into cycles, and each connection is allocated a fraction of each cycle. In this case, each connection is isolated; a delay guarantee for a connection can be calculated based on only the traffic of that connection and the fraction of allocated bandwidth. In this section, we discuss both rate-based and delay-based schedulers and review available delay guarantees for both scheduler classes. We evaluate a scheduling discipline based on the delay guarantees it can provide as well as overhead costs required for its implementation.

3.2.1 Rate-based Scheduling Disciplines

All rate-based scheduling disciplines emulate one of two systems: (1) a Time-Division-Multiplexing (TDM) system that divides time into fixed-sized frames which are in turn divided of time slots, allocating these slots to connections, or (2) a Generalized-Processor-Sharing (GPS) system that allocates a service share to each connection and provides service to each connection in proportion to its share.

The Stop-and-Go [41], Hierarchical-Round-Robin (HRR) [52], and Virtual Clock (VC) [116] schedulers approximate TDM systems which isolate connections from one another, guaranteeing a fixed share of available bandwidth to each connection. Stop-and-Go and HRR use

[2] We note that delay-based and rate-based disciplines can provide guarantees on both maximum delay and minimum throughput; the crucial difference is that while delay guarantees of a delay-based discipline are calculated directly, the delay guarantees of a rate-based discipline are derived indirectly from its throughput guarantees.

framing schemes distinguished from TDM in that (a) packets arriving during a frame may be queued for transmission during later frames and (b) departing packets from different connections are ordered arbitrarily rather than cyclically within a frame. A disadvantage of these framing disciplines is that (similar to TDM) bandwidth is unused if connections do not submit traffic at a constant rate.

The VC scheduler takes a different approach as it abandons the framing strategy and instead statistically multiplexes packets in an order that approximates TDM. A packet arriving to a VC scheduler is assigned a virtual finishing time that corresponds with the time the packet would depart a TDM system, and queued packets are transmitted in increasing order of virtual finishing times. As compared to the framing schemes, VC has the advantage of utilizing spare bandwidth that would otherwise be wasted by bursty connections. However, overhead costs of implementing VC are much higher than for the framing schemes since packets must be sorted according to their virtual finishing times.

A packet scheduler similar to VC that has recently received much attention is Weighted-Fair-Queueing (WFQ) [25]. Like VC, WFQ statistically multiplexes packets according to virtual finishing times, but WFQ emulates a fluid-flow processor-sharing system rather than TDM. WFQ is popular because it can support delay guarantees identical to those for VC schedulers [82] while simultaneously providing *fairness* properties well-suited to bursty connections that do not require delay guarantees. In particular, we note that the VC scheduler may penalize a connection for using spare bandwidth [116]; the WFQ scheduler addresses this problem by employing a more sophisticated (but computationally expensive) method for assigning virtual finishing times.

In the remainder of this section we the discuss three categories of rate-based scheduling disciplines: (1) framing disciplines, (2) VC and its variants, and (3) WFQ and its variants.

3.2.1.1 Framing Disciplines

The Stop-and-Go and HRR disciplines both use framing strategies to allocate bandwidth to connections. These disciplines divide time into fixed-size frames of size T, and connections are each allocated a fixed fraction of each frame. We assume without loss of generality that frames are aligned such that time 0 occurs at a frame boundary, that is, the frames partition time into intervals $[(i-1)\cdot T, i\cdot T]$, where i is an integer, and we refer to the frame $[(i-1)\cdot T, i\cdot T]$ as the ith frame.

Stop-and-Go defines both *arriving frames* and *departing frames*, and all packets arriving during the ith arriving frame are transmitted during the $(i+1)$th departing frame. The transmission order of packets in a departing frame is arbitrary. Delay and delay jitter guarantees for Stop-and-Go are derived for connections with traffic that conforms to the (r, T)-model: both the delay and delay jitter for all packets are bounded by $2T$ [41].

The HRR scheduler can be viewed as an extension of a simple round robin server. HRR maintains multiple service lists which are labeled $1, 2, \ldots, n$, and service list i has a frame length T_i with $T_i < T_j$ for $i < j$. Each connection is allocated some number of service cycles in one of the service lists, and HRR will cycle through service list i every T_i time units. HRR interleaves the transmission of packets from different service lists. By using multiple service lists, the HRR scheduler is able to provide a range of transmission rates; a smaller frame time yields more frequent service and hence a higher transmission rate for its connections. For a connection in service list i that conforms to the (r, T_i)-model, the maximum delay experienced by a connection is limited to $2T_i$.

Stop-and-Go and HRR share several significant disadvantages. First, both scheduling disciplines are non-work conserving since unused bandwidth cannot be utilized. In Stop-and-Go, packets that arrive during the current arriving frame cannot be transmitted until the next departing frame occurs. Similarly, the HRR scheduler will idle if it cycles to a connection that does not have a packet waiting for transmission. Also, we note that the bandwidth allocation is coupled with the delay guarantee for both packet schedulers. To address this problem, the schemes described in [41, 52] propose a hierarchical structure with multiple frame lengths that all divide the frame size T, but coupling still persists [110].

3.2.1.2 Virtual Clock (VC)

The VC scheduler statistically multiplexes packets in an order that approximates a TDM system. Each packet arriving to the VC scheduler is assigned a virtual finishing time based on the time the packet would depart a TDM system, and packets are ordered according to their virtual finishing times.

Each connection i has an associated transmission rate ρ_i, and the kth packet on connection i is assigned a virtual finishing time F_i^k. When the kth packet on connection i arrives to the scheduler with transmission time s at time t, it is assigned a virtual finishing time as follows:

$$F_i^k \leftarrow \max\{t, F_i^{k-1}\} + \frac{s}{\rho_i} \tag{3.7}$$

Observe in equation (3.7) that F_i^k is set equal to the finishing time of the packet on connection i if packets are transmitted at fixed rate ρ_i.[3] The arriving packet is inserted into the sorted transmission queue according to F_i^k. Delay guarantees for VC schedulers were developed in [32, 107] for connections that conform to the (σ, ρ)-model.

Two variants of the VC scheduling discipline are Burst Scheduling [63] and Leave-in-Time [33]. In Burst Scheduling, traffic is not assumed to be a sequence of packet arrivals but rather a sequence of packet bursts. While VC assigns a virtual finishing time to each individual packet, all packets in the same burst receive the same virtual finishing time in a Burst Scheduler. Bursts of packets could be, for example, all packets that comprise a single frame of video. The idea of scheduling packets in bursts is expanded to other scheduling disciplines in [64]. The Leave-in-Time discipline is identical to VC, but arriving packets may be delayed in a *rate-controlling mechanism* before they are inserted into the transmission queue. Rate-controlling mechanisms are included for each connection at a

[3] F_i^0 is initialized to the wall clock time upon the arrival of the first packet on connection i.

switch to ensure that the burstiness of the packet stream does not increase at downstream nodes. By controlling traffic distortions, end-to-end guarantees on delay jitter guarantees can be easily enforced [109]. We would like to note, however, that the notion of combining rate-controlling mechanisms with packet schedulers is not unique to VC; the addition of rate-controlling mechanisms has enabled other disciplines to provide delay and delay jitter guarantees [31, 102, 108, 111].

A drawback of VC scheduling is that a connection may be penalized for utilizing spare bandwidth that it received when other connections were idle [82, 116]. The scheduling disciplines discussed in the next section address this problem.

3.2.1.3 Weighted Fair Queueing (WFQ)

The WFQ scheduler, also known as Packet-by-packet Generalized Processor Sharing (PGPS), approximates a GPS system which is described in the following. In a GPS system with a set $\mathcal{N}$ of connections, each connection $i \in \mathcal{N}$ is assigned a service share ϕ_i, and the service rate provided to a connection i with waiting packets at time t is $\frac{\phi_i}{\sum_{j \in B(t)} \phi_j}$, where $B(t) \subseteq \mathcal{N}$ represents the set of backlogged connections (i.e., those connections with waiting packets) at time t. Thus, the bandwidth allocations to all connections are proportional to their service shares. Note that in the worst case, that is, when all $B(t) = \mathcal{N}$, each connection receives a minimum guaranteed rate of $g_i = \frac{\phi_i}{\sum_{j \in \mathcal{N}} \phi_j}$. However, GPS is impractical to implement since it does not transmit packets as entities but rather requires bit-by-bit multiplexing. To address this problem, packetized versions of GPS have been considered.

WFQ approximates GPS in the same way that VC approximates TDM: packets are assigned virtual finishing times corresponding with the time they would complete transmission in a GPS system [25, 82]. Each connection is assigned a service share, and the kth packet on connection i that has transmission time s and arrival time t is assigned the following virtual finishing time:

$$F_i^k \leftarrow \max\{t, F_i^{k-1}\} + \frac{s \cdot \sum_{j \in B(t)} \phi_j}{\phi_i} \tag{3.8}$$

By comparing equation (3.8) with that for VC in equation (3.7), we note that computation of virtual finishing times in WFQ depends on the set of backlogged connections. WFQ simulates a GPS system and orders packets according to their departure times in the simulated system. Parekh and Gallagher analyze the degree to which WFQ and PGPS approximate the fairness of GPS in [82]. They show that a packet which departs a GPS scheduler at time t will depart a WFQ scheduler no later than time $t + s^{max}$, where s^{max} is the maximum transmission time of any packet in the system. Thus, a WFQ scheduler will never fall behind a corresponding GPS scheduler by more than a single packet transmission time.

Variants of WFQ include the Worst-case Fair Weighted Fair Queueing (WF^2Q) and Self-Clocked Fair Queueing (SCFQ) schedulers [9, 42]. A WF^2Q scheduler sorts packets according to their finishing time in a GPS scheduler, but WF^2Q is a non-work-conserving scheduler in that it will not begin transmission of a packet until its *eligibility time* which is defined to be the time the packet starts service in the corresponding GPS system. Bennett and Zhang show in [9] that a packet from connection i departing a GPS scheduler at time t will depart a WF^2Q scheduler no later than time $t + s^{max}$ and no earlier than time $t - g_i \cdot s^{max}$. Thus, WF^2Q has stronger fairness properties than WFQ at the expense of increased implementation complexity.

In contrast to WF^2Q, the SCFQ scheduler proposed by Golestani in [42] sacrifices fairness for significantly reduced implementation complexity. In real time, both WFQ and WF^2Q must simulate a GPS scheduler to calculate a precise sorting criteria, and this task is computationally expensive. The SCFQ scheduler instead computes a sorting criteria that uses using the progress of its own scheduler as a reference rather than that of a simulated GPS scheduler. While a SCFQ scheduler is the simplest of all GPS approximation schedulers described here, it has the worst fairness properties as well [9, 43].

Other packet schedulers that approximate WFQ include Deficit Round Robin (DRR) [96], Carry-Over Round Robin (CORR) [91], and Frame-based Fair Queueing (FFQ) [98]. The two round-robin schedulers are work-conserving and allocate unused bandwidth to backlogged connections based on their service shares. Although these schedulers can be implemented much more efficiently than WFQ, they do not provide delay guarantees or fairness properties that compete with WFQ. Frame-based Fair Queueing (FFQ) [98] is a tunable scheduling discipline that provides a tradeoff between the implementation simplicity of VC and the fairness properties of WFQ.

3.2.2 Delay-based Scheduling Disciplines

In this section we review two delay-based scheduling disciplines that have been previously considered for bounded-delay services: *Earliest-Deadline-First* (EDF) [31, 102] and *Static-Priority* [111]. Similar to our discussion of the rate-based schedulers, we describe the operations of the schedulers. However, here we also review delay bound tests that can be used in admission control tests; similar tests are not available for the rate-based schedulers.

3.2.2.1 Earliest-Deadline-First (EDF)

An EDF scheduler assigns each arriving packet a timestamp corresponding to its deadline, i.e., a packet from connection j with a delay bound d_j that arrives at the scheduler at time t is assigned a deadline of $t + d_j$. The EDF scheduling algorithm always selects the packet with the earliest deadline for transmission, and it thus maintains a single queue of untransmitted packets sorted in increasing order of packet deadlines. The scheduler always selects the packet in the first position of the queue, that is, the packet with the lowest deadline, for transmission; however, the transmission of a packet is not interrupted by the arrival of a packet with a lower deadline. Since the scheduler queue of an EDF-scheduler must be sorted according to deadlines, each packet arrival involves a search operation to find the correct position of the newly arrived packet in the scheduler queue.

The EDF scheduler achieves optimal network utilization in the following sense: if any packet scheduler can support a set of connections with delay constraints, then so can EDF. This optimality was demonstrated in a network setting for traffic conforming to the (σ, ρ)-

model in [38] and for more general traffic characterizations in [74]. The high achievable network utilization makes EDF an excellent candidate for bounded-delay services.

Ferrari and Verma presented sufficient schedulability conditions for EDF scheduling for a bounded delay service in [31]. Using a traffic specification which neglects the burstiness of network traffic, Zheng and Shin have derived necessary and sufficient schedulability conditions [118]. Liebeherr et. al. present necessary and sufficient conditions for schedulability in an EDF scheduler in [74] for the general class of traffic characterizations described in Section 2.1. We assume without loss of generality that connections are ordered so that $i < j$ whenever $d_i < d_j$. Then the schedulability conditions are given as follows for a set of connections $\mathcal{C}$ [74]:

Theorem 3.1 *A set $\mathcal{C}$ of connections given by $\{(A_j^*, d_j)\}_{j \in \mathcal{C}}$ is EDF-schedulable if and only if for all $t \geq d_1$:*

$$t \geq \sum_{j \in \mathcal{C}} A_j^*(t - d_j) + \max_{d_k > t} s_k \tag{3.9}$$

This condition can be considerably simplified for the practical traffic models described in Section 3.1 [74].

3.2.2.2 Static-Priority (SP)

An SP-scheduler partitions the set $\mathcal{C}$ of connections into P connection sets $\{\mathcal{C}_p\}_{1 \leq p \leq P}$, where all connections in set $\mathcal{C}_p$ have the same delay bound d_p, with $d_p < d_q$ for $p < q$. SP maintains a set of P prioritized FIFO queues, labeled FIFO 1, FIFO 2, ..., FIFO P, where FIFO p is associated with connection set $\mathcal{C}_p$ and a smaller index indicates a higher priority. Thus, the priority of a connection is high if its delay bound is short. All packets arriving on a connection j in connection set $\mathcal{C}_p$ are inserted into FIFO p. At the beginning of a busy period, or after completing the transmission of a packet, the SP-scheduler always selects the first packet in the nonempty FIFO queue with the highest priority for transmission.

Since the SP scheduler does not maintain a sorted list of untransmitted traffic as the EDF, VC, and WFQ schedulers do, the scheduling operations of an SP-scheduler involve fewer overhead computations. Due to its simplicity which enables packet scheduling at very high data rates, SP schedulers are attractive for bounded delay services.

Using a fluid flow traffic specification, necessary and sufficient schedulability conditions for SP-schedulers are presented in [23]. However, the conditions are not exact for more realistic discrete traffic scenarios. For a particular discrete traffic specification [31], Zhang and Ferrari [111], and Zhang [108] have derived several sufficient schedulability conditions. The following necessary and sufficient schedulability conditions for SP schedulers are presented in [74]:

Theorem 3.2 *A set $\mathcal{C}$ of connections given by $\{(A_j^*, d_j)\}_{j\in\mathcal{C}}$ is SP-schedulable if and only if for all priorities p and for all $t \geq 0$ there exists a τ with $\tau \leq d_p - s_p^{min}$ such that:*

$$t + \tau \geq \sum_{j\in\mathcal{C}_p} A_j^*(t) + \sum_{q=1}^{p-1} \sum_{j\in\mathcal{C}_q} A_j^*((t+\tau)^-) - s_p^{min} + \max_{r>p} s_r \tag{3.10}$$

Comparing Theorems 3.1 and 3.2, we see that testing (exact) schedulability for SP schedulers requires significantly more effort than for EDF schedulers. First, condition (3.10) must be tested for each priority level. Second, for a fixed priority p and fixed value of t, condition (3.10) must possibly be tested for the entire range of values of τ. Similar to EDF, the conditions can be considerably simplified for the traffic models described in Section 3.1 [74].

3.2.3 Discussion of Tradeoffs

A packet scheduling discipline for a bounded-delay service should satisfy both of the following criteria:

- The scheduling discipline should be able to support a diverse set of connections with different traffic characterizations and delay constraints while maintaining high network utilization.

- The scheduling discipline should only require modest overhead requirements so that it can be implemented in a high-speed network.

Here we discuss the degree to which both the rate-based and delay-based packet schedulers satisfy these criteria.

With regard to achievable network utilization, EDF is the a better choice than any of the rate-based disciplines for a number of reasons. First, recall that EDF is the optimal packet scheduler for a single network switch; also, in [39] it is shown that EDF can achieve a utilization higher than both WFQ and VC in a network environment over multiple switches. Second, delay guarantees for rate-based packet schedulers are only available for a restricted set of traffic models; note that the condition for EDF in Theorem 3.1 applies to all traffic models reviewed in Section 3.1. Finally, the delay guarantees for rate-based packet schedulers are not in the form of delay bound tests that can be easily applied to admission control mechanisms but are rather computed as a maximum delay bound based on throughput guarantees. For these reasons, EDF is superior to the rate-based disciplines for networks with a bounded-delay service.

Turning to implementation overhead, EDF, WFQ, and VC all require the maintenance of a sorted transmission queue, and the WFQ scheduler is the most difficult of the three to implement since it must simulate a fluid-flow GPS system. However, the SP scheduling discipline can be implemented much more efficiently than any of these schemes since it does not require sorting operations. We discern an inherent tradeoff between the high utilization of EDF and the simple implementation of SP; we will design a scheduling discipline that takes advantage of this tradeoff in Chapter 5.

4

Fast Video Traffic Characterization for QoS Networks

A key component of a resource reservation scheme is the traffic characterization used to specify the traffic arrivals on a video connection. A bounded-delay service requires a deterministic traffic characterization that provides an upper bound on traffic arrivals. It is important to specify the traffic on a connection as accurately as possible since the traffic characterization will be used in admission control mechanisms that verify if sufficient resources are available within the network to support the traffic on the connection at the desired QoS. If the traffic characterization is too pessimistic in describing the traffic, the admission control mechanisms will overestimate the resource requirements of a connection, resulting in poor network utilization. Due to the complex timely correlations of VBR video sequences, elaborate traffic characterizations have been devised which achieve a high degree of accuracy [34, 37, 58, 87, 88].

While admission control mechanisms require that traffic characterizations are accurate in describing the worst-case traffic, traffic policing mechanisms that monitor in real time if the traffic submitted to the network conforms to its traffic characterization require a simple traffic characterization [86]. Therefore, the choice of traffic characterization method is a tradeoff between the high accuracy preferred by admission control mechanisms and the simplicity required for implementing traffic policing mechanisms.

As we discussed in Chapter 3, the $(\vec{\sigma}, \vec{\rho})$-model that can be policed with a fixed number of leaky bucket policing mechanisms satisfies both criteria to a high degree. Since a leaky bucket can be implemented with a single counter and a single timer [86], leaky buckets seem to satisfy the need for simple traffic characterizations. In a previous study [103], we showed that concave piecewise-linear functions ("leaky buckets") are capable of accurately characterizing VBR video traffic. However, the number of leaky buckets needed for an accurate characterization was shown to be large. For example, up to a dozen leaky buckets are needed to accurately characterize MPEG-I video streams [103]. Since practical considerations limit the number of available leaky buckets to a small value, e.g., the limit is two for ATM connections [1], methods are needed that yield an accurate VBR traffic characterization, yet, with only a small number of leaky buckets.

In this chapter, we present a solution to the problem of constructing an accurate traffic characterization for stored VBR video with few leaky buckets. Our solution approach is based on the so-called *empirical envelope* of a video sequence described in Chapter 1 [103]. As we will show in Section 4.2, the empirical envelope for a connection is the most accurate deterministic traffic characterization. However, recall that the empirical envelope itself is not practical for use in QoS networks for two reasons: (1) the empirical envelope requires a large number of parameters which are computationally expensive to produce, and (2) the traffic specified by the empirical envelope cannot be policed using simple traffic policing mechanisms. The characterization method presented in this chapter addresses both of these problems. First, we determine an approximation of the empirical envelope based on a subset of its parameters that can be computed quickly. We then use this approximation to determine a traffic characterization that can be policed by a small number of leaky buckets.

We demonstrate the effectiveness of our method in networks with a bounded-delay service using traffic traces of two 25-30 minute MPEG encoded video segments [35]. Our examples illustrate the minimum number of empirical envelope parameters and leaky bucket mechanisms needed to obtain an accurate traffic characterization. We show that only 200 out of a total 40,000 envelope parameters and three leaky bucket mechanisms are sufficient to produce traffic characterizations leading to utilizations within 91% of the results achievable with the empirical envelope. In a case study, we show how our methods can be employed in networks with dynamic resource reservation schemes, i.e., where the traffic characterization can be renegotiated after the connection is established. We demonstrate that a renegotiation scheme can yield increases in network utilization of 20-35%. The fast characterization method developed in this chapter is well-suited to dynamic reservation schemes since renegotiation requires the calculation of multiple traffic characterizations.

The remainder of this chapter is structured as follows. In Section 4.1 we discuss previous work on selecting traffic parameters for traffic characterizations that conform to leaky bucket mechanisms. We present our traffic characterization method in Sections 4.2 and 4.3; In Section 4.2 we describe a method for approximating the empirical envelope using only a small number of envelope parameters, and in Section 4.3 we describe an algorithm for selecting leaky bucket parameters. In Section 4.4 we present a case study where we apply our method to construct a renegotiation scheme for a bounded-delay service.

4.1 Related Work

Several studies have considered deterministic traffic characterizations for VBR video traffic using the (σ, ρ) traffic model that corresponds to the leaky bucket policing mechanism. Most studies use only a single (σ, ρ) pair and explore the dependencies between the burstiness parameter σ and the rate parameter ρ [77, 80, 87, 88, 90]. In particular, for any fixed choice of rate ρ, the burst parameter σ should be selected as small as possible, that is [77]:

$$\sigma = \inf\{\hat{\sigma} \mid \hat{\sigma} + \rho t \geq A[\tau, \tau + t],\ \forall t,\ \tau \geq 0\} \tag{4.1}$$

Equation (4.1) illustrates a tradeoff between buffer space (i.e., burst) and bandwidth (i.e., rate) when selecting leaky bucket parameters. By combining the dependency in equation (4.1) with all rates ρ between the average and peak rate of a connection, one obtains an infinite candidate set of (σ, ρ) pairs from which all leaky bucket parameters should be selected. Note that it is computationally demanding to determine this candidate set of (σ, ρ) pairs.

Many schemes select parameters σ and ρ according to either network resource availability or the relative importance of bandwidth and buffer space. Pancha and El Zarki [80] choose parameters by fixing the burstiness parameter σ according to available buffer space, while the choice of (σ, ρ) in [18] depends on the relative availability of unallocated bandwidth and buffer space. An approach discussed by Guillemin et. al. in [46] assigns relative importance parameters α and β to buffer space and bandwidth, respectively; the pair (σ, ρ) is selected to minimize the quantity $\sigma^{\alpha} \cdot \rho^{\beta}$. The authors note that a "natural" choice is the case where both resources have the same cost, that is, $\alpha = \beta = 1$. A drawback of all of these methods is that they do not strive for high network utilization as a design goal. Also, all of these approaches consider the selection of parameters for only a single leaky bucket mechanism.

Guillemin et. al. present two heuristic algorithms in [46] that select a leaky bucket pair (σ, ρ) to approximate an "ideal" probabilistic traffic characterization, the so-called *time-ϵ quantile function* $M_\epsilon(t)$ associated with a source. The heuristic algorithms are similar to the characterization method proposed in this chapter in that they first determine a function that describes the traffic on a connection and then determine parameters based on this function. Assuming that N_t is a random variable specifying the number of packets generated over any interval of length t, a function $M_\epsilon(t)$ is used to specify with probability $1 - \epsilon$ the maximum traffic arrivals n in any interval of length t [46, 90]:

$$M_\epsilon(t) = \inf\{n,\ Pr\{N_t \geq n\} \leq \epsilon\} \tag{4.2}$$

The quantity $M_\epsilon(t)/t$ specifies the rate of the video sequence over multiple time scales t. The first heuristic in [46] selects a leaky bucket parameter $(\sigma + \rho t)$ such that the maximum difference between $M_\epsilon(t)/t$ and the "normalized" leaky bucket curve $(\sigma + \rho t)/t$ is minimized. The second heuristic minimizes the area *y between* the normalized curves $M_\epsilon(t)/t$ and $(\sigma + \rho t)/t$ over an interval $[0, T_0]$. As noted in [46], the selection of parameters for the second heuristic is heavily dependent on the choice of T_0 which is not set explicitly in the paper.

While the focus of our research is on finding a traffic characterization for VBR video traffic, other studies exist that explore the benefits of reducing the burstiness of VBR traffic through either (1) *shaping* the traffic by spacing packets before submitting them to the network [39, 56, 62] or (2) sending packets early with respect to their playback time at the receiving application via *workahead smoothing* [78, 88, 92]. These techniques involve modification of the traffic A that is submitted to the network on a connection by buffering at either the sender, receiver, or a combination of both. While shaping and smoothing techniques have been shown to increase the achievable network utilization, these methods are orthogonal to the traffic characterization problem studied in this research. Note that even after traffic is shaped or smoothed, a characterization method such as the one developed here must be available to determine an accurate and policeable characterization for the traffic submitted to the network.

4.2 A Fast Characterization Method for VBR Video

None of the the characterization approaches for VBR video with leaky buckets described above attempt to maximize the number of admissible connections in a QoS network. Wrege, Knightly, Zhang, and Liebeherr present in [103] a traffic characterization, referred to as the "empirical envelope", that maximizes resource utilization. They show how to approximate the empirical envelope with leaky buckets, however, the number of parameters of the resulting traffic constraint function is considerable: up to 12 leaky buckets were needed for an accurate characterization of an MPEG video sequence [103]. Also, the computational complexity of the characterization algorithms was substantial.

Here we present a method to obtain VBR video traffic characterizations that can be policed by a small, fixed number of leaky buckets. The computational complexity our our method is low and efficient as compared to the methods in [46, 103].

In Subsection 3.1 we discuss the tradeoffs of traffic characterization methods that are based on the empirical envelope. Following, in Subsections 3.2 and 3.3, we present and evaluate the new solution approach to VBR video characterization.

4.2.1 The Empirical Envelope E^*

The tightest traffic constraint function for a given traffic source is its *empirical envelope*, denoted by E^* [17, 103]. The empirical envelope E^* of a video sequence is optimal in the sense that, for any subadditive traffic constraint function A^* that satisfies equation (2.1), $A^*(t) \geq E^*(t)$ for all t. The empirical envelope E^* is given by the following equation [17, 103]:

$$E^*(t) = \sup_{\tau \geq 0} A[\tau, \tau + t] \qquad \forall t \geq 0 \tag{4.3}$$

Note from equation (4.3) that E^* is subadditive.

The following method presented in [103] obtains the empirical envelope of a given video sequence consisting of N frames with fixed inter-frame time r. We assume that frames are fragmented into 53-byte ATM cells with a payload of 48 bytes each, and these cells are transmitted at equally-spaced intervals over the frame time r. If the sequence of frame sizes of a video sequence is given by $\{f_1, f_2, \ldots, f_N\}$, then the empirical envelope E^* can be constructed by calculating [103]:

$$E^*(ir) = \max_{0<k<N-i+1} \sum_{j=k}^{k+i-1} f_j \qquad \text{for } i = 1, 2, \ldots N \tag{4.4}$$

Note that equation (4.4) defines N parameters $\{E^*(ir) \mid 1 \leq i \leq N\}$ for the empirical envelope, where $E^*(r)$ is equal to the largest frame in the video sequence, $E^*(2r)$ is equal to the largest two-frame sequence, etc. The values of the empirical envelope at times that are not multiples of the frame time are obtained by spacing the cells in $E^*(ir) - E^*((i-1)r)$ evenly over the frame time $[(i-1)r, ir]$.

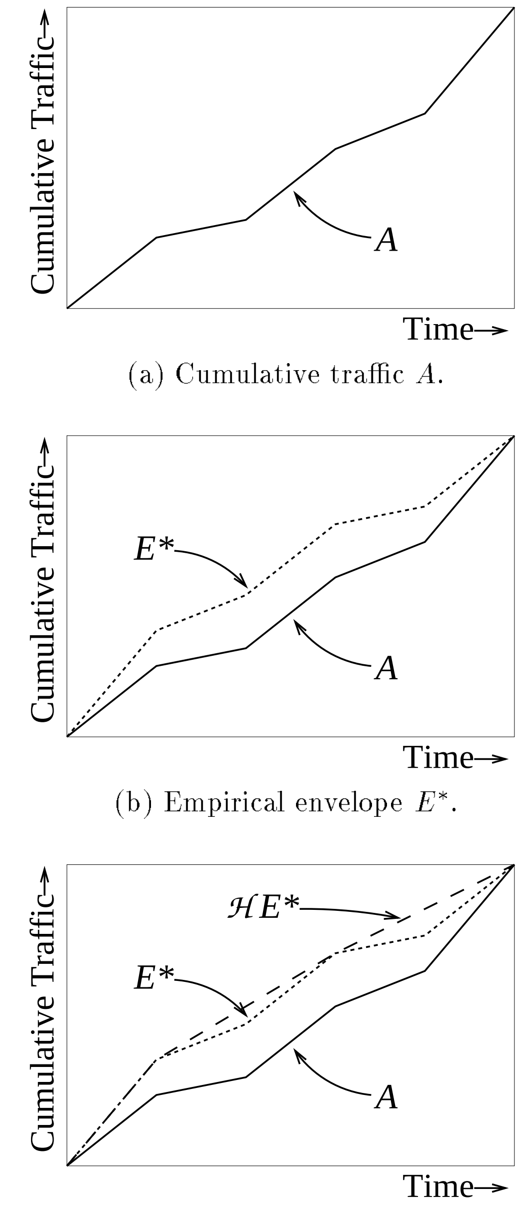
Since the empirical envelope E^* does not conform to a parameterized traffic model, it is difficult to police. In previous work, we showed how to determine a $(\vec{\sigma}, \vec{\rho})$-model traffic characterization based on the *concave hull of* E^*, which we denote by $\mathcal{H}E^*$ [103].[1] Since the function $\mathcal{H}E^*$ is the smallest piecewise-linear concave function larger than E^* [21], $\mathcal{H}E^*$ is most accurate traffic characterization that can be policed by leaky buckets.

In Figure 4.1 we illustrate the traffic characterization method from [103] with an example. The cumulative traffic arrivals A for a traffic source are depicted in Figure 4.1(a). Figures 4.1(b) and 4.1(c) show the empirical envelope E^* and the concave hull $\mathcal{H}E^*$, respectively, for this traffic source. The relationship between A, E^*, and $\mathcal{H}E^*$ for an actual MPEG-encoded video sequence is illustrated in Figure 4.2. Figure 4.2(a) shows a trace of 250 frames of an MPEG movie. The traffic is packetized into ATM cells with 48-byte payloads, and we plot the number of cells as a function of the frame sequence number. In Figure 4.2(b), we illustrate the cumulative cells A for the trace in Figure 4.2(a), and we also plot the empirical envelope E^* and its concave hull $\mathcal{H}E^*$.

4.2.2 Approximating the Envelope with Extrapolations

The traffic characterization method outlined in the previous subsection was shown [103] to produce very accurate traffic characterizations based on the empirical envelope. However, the empirical envelope requires a large number of parameters, that is, one parameter per frame in the sequence. The number of operations required to compute all N parameters of the empirical envelope E^* for a video sequence with N frames is $O(N^2)$. Since N is generally large, e.g., it exceeds 200,000 for most feature-length motion pictures, it may not be possible to calculate the empirical envelope in real-time. Note that while the characterization $\mathcal{H}E^*$

[1] In this chapter, we use $\mathcal{H}$ to denote the concave hull operator, that is, $\mathcal{H}f$ is the concave hull of the function f.

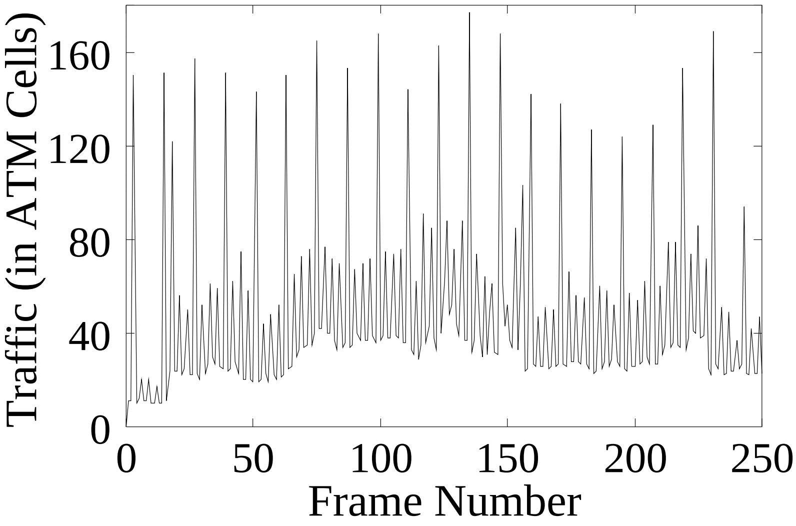


(a) Cumulative traffic A.

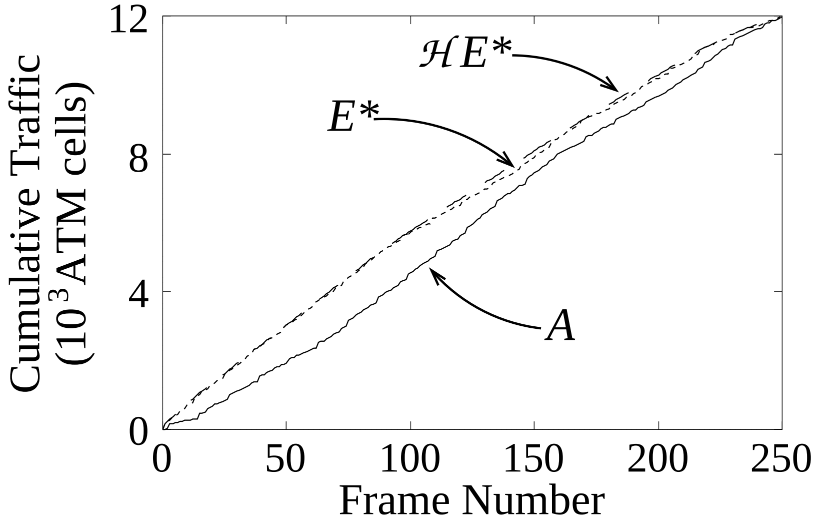
(b) Empirical envelope E^*.

(c) Concave approximation $\mathcal{H}E^*$.

Figure 4.1: Characterization approach using the empirical envelope [103].



(a) MPEG traffic trace.



(b) Cumulative traffic and constraint functions.

Figure 4.2: Functions A, E^* and $\mathcal{H}E^*$ for an actual MPEG trace.

uses fewer parameters, determining $\mathcal{H}E^*$ requires knowledge of the entire empirical envelope, and so its production is also computationally expensive. We therefore seek other traffic constraint functions that closely approximate the envelope but can be calculated with fewer parameters.

Here we present two methods for obtaining viable traffic constraint functions defined for all times t that are derived only from the first k parameters of the empirical envelope, i.e., $E^*(r), E^*(2r), \ldots, E^*(kr)$. Both methods construct a traffic constraint function through extrapolation of these k parameters. We first discuss the best-possible extrapolation based on the first k parameters of E^* and then present a simple characterization that can obtained with a fast extrapolation technique.

Any viable traffic constraint function obtained from the first k parameters of the envelope must be at least as large as the empirical envelope E^* for all times t. Since we know that E^* is a subadditive function, the best extrapolation is given by the *largest subadditive extrapolation of* $\{E^*(ir)\}_{1\leq i\leq k}$. We denote this largest subadditive extrapolation by E_k^*, where E_k^* is obtained by calculating:

$$E_k^*(ir) = \begin{cases} E^*(ir) & \text{for } i \leq k \\ \min\limits_{1\leq j<i} \{E_k^*(jr) + E_k^*((i-j)r)\} & \text{for } i > k \end{cases} \tag{4.5}$$

E_k^* is equal to the empirical envelope for the first k frame times, and E_k^* is defined for subsequent times by exploiting the requirement for subadditivity of E_k^*.[2]

Although the function E_k^* is the tightest traffic constraint function that can be obtained directly from the first k parameters of the envelope, the production of E_k^* requires a large number of computations. Specifically, we see from equation (4.5) that the number of computations required to construct E_k^* is $O(N^2)$, the same number required for computing the empirical envelope itself. Since we seek an approximation that can be computed efficiently,

[2]Note that equation (4.5) only defines E_k^* for times that are multiples of the frame time r. Similar to the production of the empirical envelope in equation (4.4), the values for intermediate values of E_k^* are determined by spacing cells evenly over each frame.

we turn to other approximation schemes, and we will use E_k^* as a benchmark for other approximations.

As a more efficient extrapolation, we next consider a function that is obtained by simply repeating the first k parameters $\{E^*(ir)\}_{1\leq i\leq k}$ for all times t. We call such a function the *repetition extrapolation*, which we denote by R_k^*. R_k^* is given as follows:

$$R_k^*(t) = \lfloor \frac{t}{kr} \rfloor E^*(kr) + E^*(t - \lfloor \frac{t}{kr} \rfloor (kr)) \quad \text{for } t \geq 0 \tag{4.6}$$

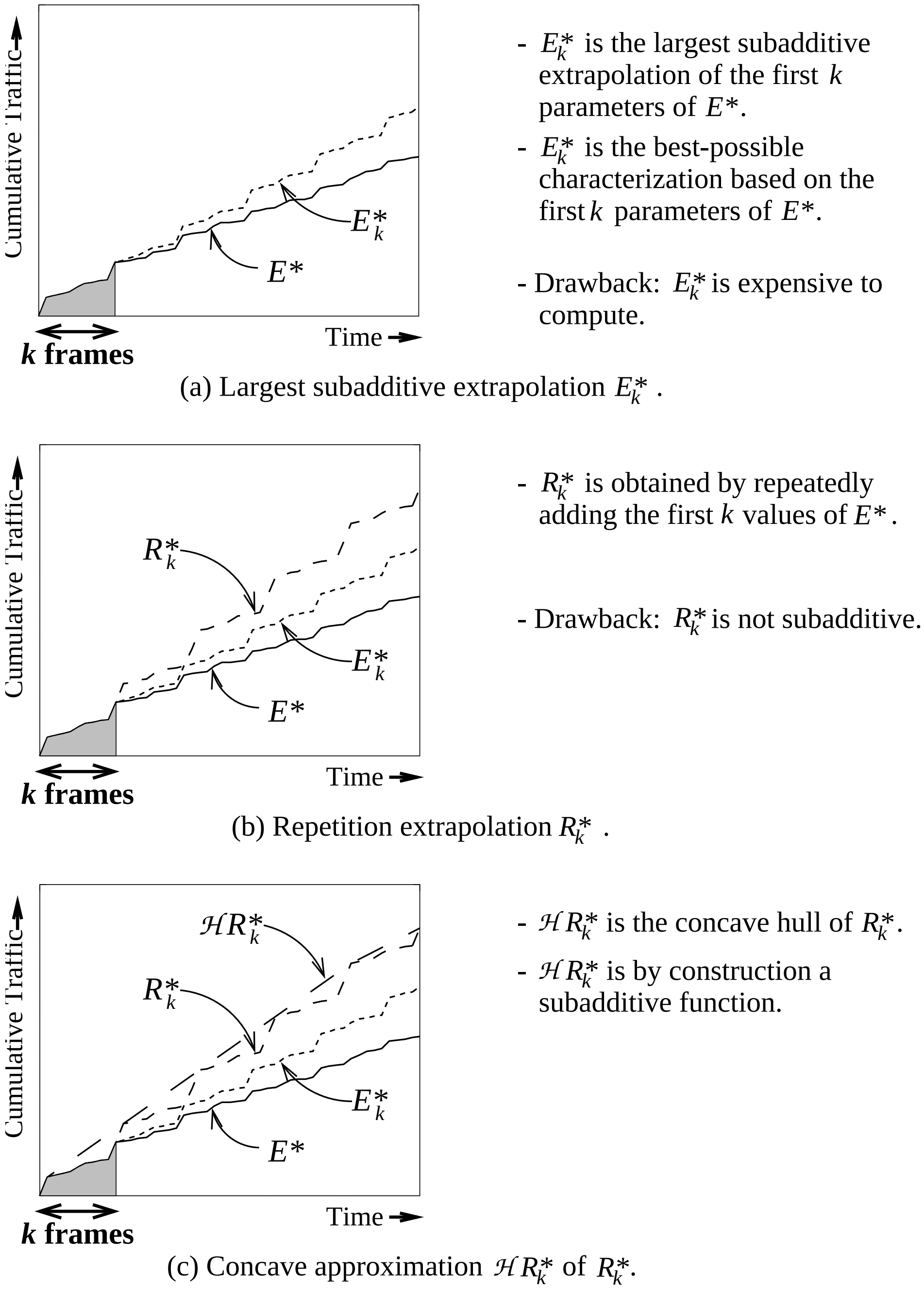
Observe that R_k^* can be immediately obtained from the first k parameters of the envelope, and so the computational complexity of computing R_k^* is $O(kN)$. For small values of k, R_k^* can be computed much more efficiently than the entire empirical envelope E^*.

Although R_k^* provides a time-invariant bound on the traffic arrivals A in terms of equation (2.1), it is not necessarily subadditive and hence does not satisfy our requirement for a traffic constraint function. To remedy this problem we consider yet another function $\mathcal{H}R_k^*$, the *concave hull of* R_k. The concave hull $\mathcal{H}R_k^*$ is by construction a viable traffic constraint function since its subadditivity follows from its concavity. Note that $\mathcal{H}R_k^*$ can be expressed in terms of the $(\vec{\sigma}, \vec{\rho})$ model as follows:

$$\mathcal{H}R_k^*(t) \equiv B_n^* = \min_{1\leq i\leq n} \{\overline{\sigma}_i + \overline{\rho}_i t\}, \tag{4.7}$$

where parameters $\overline{\sigma}_i$ and $\overline{\rho}_i$ are determined by some appropriate algorithm to compute the concave hull of a function, e.g. [103].

We review the extrapolation methods in Figure 4.3. Figure 4.3(a) illustrates the relationship between the empirical envelope E^* and its approximation E_k^*, the largest subadditive extrapolation of the first k parameters of E^*. E_k^* is the most accurate traffic characterization that can be obtained from the first k values of the empirical envelope. The repetition extrapolation R_k^*, depicted in Figure 4.3(b), can be efficiently computed by repeating the first k parameters of the empirical envelope. However, R_k^* is not subadditive and therefore is not a viable traffic constraint function. The concave hull $\mathcal{H}R_k^*$, shown in Figure 4.3(c), is by construction subadditive and can be used as a deterministic traffic constraint function.



(a) Largest subadditive extrapolation E_k^*.

(b) Repetition extrapolation R_k^*.

(c) Concave approximation $\mathcal{H}R_k^*$ of R_k^*.

Figure 4.3: Approximations of the empirical envelope.

A problem that remains to be solved is the potentially large number of $(\overline{\sigma}_i, \overline{\rho}_i)$ pairs needed for the concave hull $\mathcal{H}R_k^*$, mandating a large number of leaky bucket mechanisms for a single connection. We will address this problem in Section 4.3, where we present an algorithm that approximates $\mathcal{H}R_k^*$ with a traffic characterization that can be policed by a fixed (and small) number of leaky buckets.

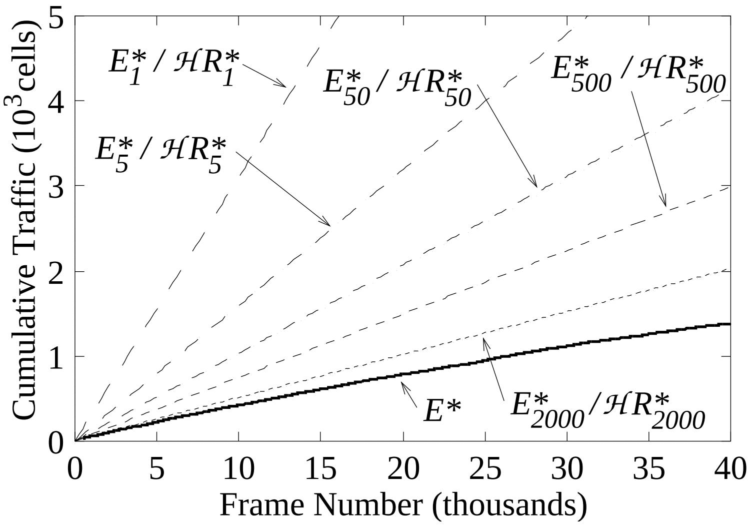
4.2.3 Evaluation

Here we evaluate the accuracy of traffic characterizations E_k^* and $\mathcal{H}R_k^*$ as approximations of the empirical envelope using actual traces of MPEG-compressed video. We are are interested in determining the size of k needed to generate an accurate characterization for a VBR video source.

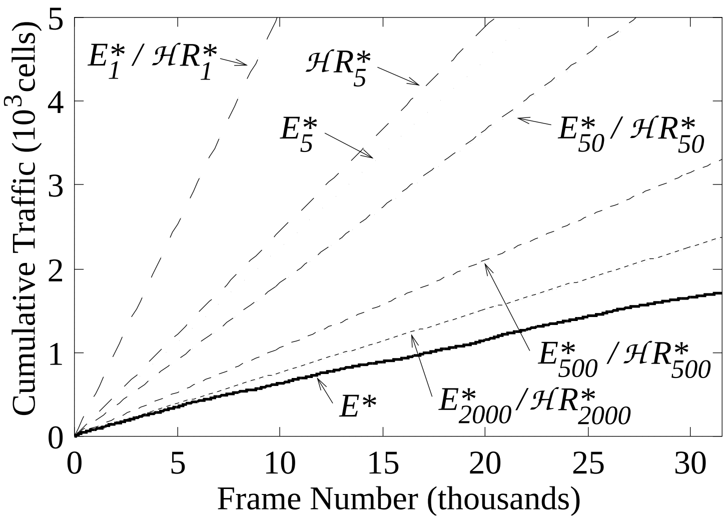
We use two MPEG traces in the evaluation: one from the entertainment film *Jurassic Park* ("*Park*"), and the second from a news broadcast ("*News*"). These sequences were encoded in software with the Berkeley MPEG-encoder [89]. Both *Park* and *News* are thirty-minute video sequences with a frame size of 384x288 and frame pattern IBBPBBPBBPBB. We note that *News* generates burstier traffic than *Park*; the ratio of the peak rate to the average rate for *News* and *Park* are 6 and 4, respectively.

Figures 4.4(a) and 4.4(b) illustrate traffic constraint functions for the *News* and *Park* traces, respectively. We show the empirical envelope E^* as well as E_k^* and $\mathcal{H}R_k^*$ for $k \in \{1, 5, 50, 500, 2000\}$. For each traffic constraint function, we plot the cumulative number of cells as a function of the frame sequence number. In both graphs, the empirical envelope E^* is shown as a bold solid curve, while the functions E_k^* and $\mathcal{H}R_k^*$ are depicted by dotted and dashed curves, respectively. As expected, the approximation functions estimate the empirical envelope E^* more accurately for larger values of k.

A key observation from Figure 4.4 is that $\mathcal{H}R_k^* \approx E_k^*$ for most values of k; only $\mathcal{H}R_5^*$ and E_5^* for the *News* sequence in Figure 4.4(b) differ considerably. Since E_k^* is the tightest traffic characterization that can be produced from k frames of the empirical envelope, we



(a) *Park*



(b) *News*

Figure 4.4: Traffic constraint functions.

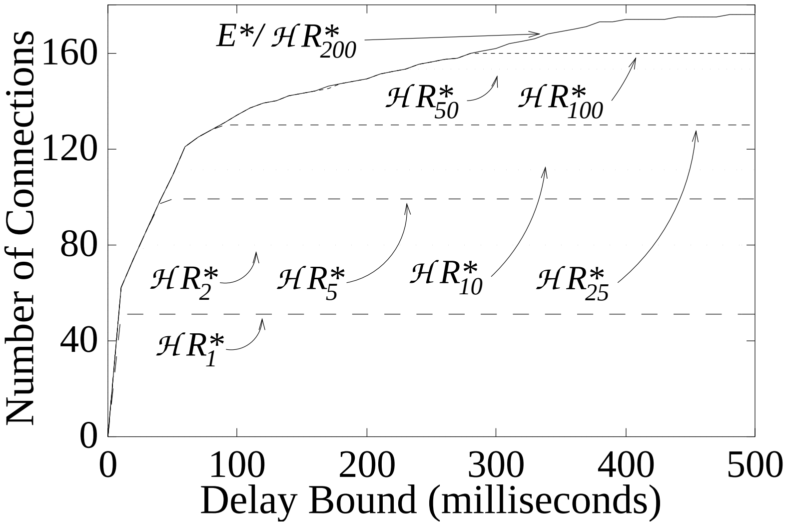
note that the concave hull of the repetition extrapolation $\mathcal{H}R_k^*$ is an accurate approximation of E_k^*, validating our selection of $\mathcal{H}R_k^*$ for the characterization.

We next consider the utilizations that can be achieved at a network switch using traffic constraint functions $\mathcal{H}R_k^*$. We assume a single multiplexer that operates at 155 Mbps, a data rate that corresponds to OC-3, and we further assume that the switch transmits its packets with a First-Come-First-Served (FCFS) discipline.[3] Figure 4.5 illustrates the network utilization obtained at a multiplexer using E^* as well as $\mathcal{H}R_k^*$ for various values of k. All connections at a multiplexer are assumed to be of the same type (either *Park* or *News*) and have identical delay bounds (in the range $0 \leq d \leq 500$ msec). For each characterization, we plot the maximum number of connections that can be admitted as a function of the delay bounds of those connections. For example, Figure 4.5(b) shows that the traffic constraint function $\mathcal{H}R_2^*$ can be used to support 61 *News* connections for delay bounds larger than 35 ms.

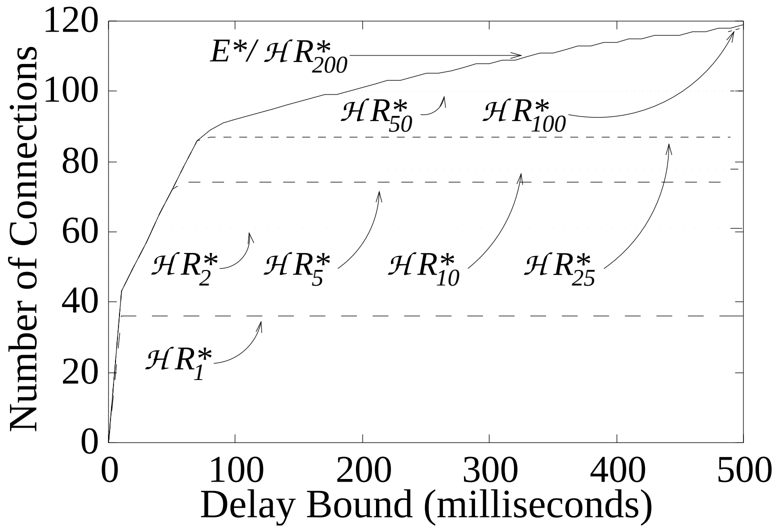
The general trend in both graphs is that the number of connections accepted using $\mathcal{H}R_k^*$ as the traffic constraint function increases with k. An important observation is that the function $\mathcal{H}R_{200}^*$ admits the same number of connections as the empirical envelope E^* for delay bounds up to 500 milliseconds. Thus, we can use our approximation function $\mathcal{H}R_k^*$ based on the first 200 parameters of the envelope (i.e., $\mathcal{H}R_{200}^*$) to characterize both video sequences for the delay bound range considered; we achieve the same utilization using $\mathcal{H}R_{200}^*$ as we would using the empirical envelope with 40,000 parameters.

However, while the function $\mathcal{H}R_k^*$ provides an accurate traffic characterization for VBR video, the number of leaky buckets required to enforce $\mathcal{H}R_k^*$ may be too large. For example, 12 leaky bucket mechanisms are needed to police $\mathcal{H}R_{200}^*$ for the *News* sequence (i.e., $\mathcal{H}R_{200}^* \equiv B_{12}^*$). Unfortunately, the number of leaky buckets available to monitor a connec-

[3] The exact delay bound test for FCFS multiplexers is given by $d \geq \sum_{j \in \mathcal{N}} A_j^*(t) - t + s$ for all $t \geq 0$ [23]. In this admission control test, $\mathcal{N}$ denotes the set of all connections at a multiplexer, d denotes the maximum delay at the multiplexer, and s denotes the transmission time of a cell.



(a) *Park*



(b) *News*

Figure 4.5: Utilization comparison.

tion in a real network is typically limited to only two or three. This problem is addressed in the next section where we present an algorithm that reduces the number of leaky buckets used to characterize a VBR video source.

4.3 Leaky Bucket Parameter Selection

At this point, we have obtained an accurate traffic characterization $\mathcal{HR}_k^*$ that conforms to the $(\vec{\sigma}, \vec{\rho})$ traffic model. We write $\mathcal{HR}_k^* \equiv B_n^*$ to indicate that n is the number of $(\overline{\sigma}_j, \overline{\rho}_j)$ pairs for the traffic characterization. Since n can be large, and since the number m of leaky buckets available to police a video connection is small, we will use a curve-fitting method that reduces B_n^* to B_m^* with $m < n$.

We formulate the problem as follows. Given a function $B_n^* = \min_{1 \leq i \leq n}\{\overline{\sigma}_j + \overline{\rho}_j t\}$, we want to find a set of $m < n$ (σ_i, ρ_i) pairs that determine a traffic constraint function B_m^*:

$$B_m^*(t) = \min_{1 \leq i \leq m} \{\sigma_i + \rho_i t\}, \tag{4.8}$$

such that $B_m^*(t) \geq B_n^*(t)$ for all t and B_m^* is a tight approximation of B_n^*. We use a cost function $C(B_m^*, B_n^*)$ to express the closeness of B_m^* to B_n^*. Assuming that we have such a cost function available, we select parameters (σ_i, ρ_i) for B_m^* as solutions to the following optimization problem:

Minimize $C(B_m^*, B_n^*)$

Subject to $B_m^*(t) \geq B_n^*(t) \;\; \forall t \geq 0.$

In the remainder of this section we describe the cost function $C(B_m^*, B_n^*)$ and present a heuristic algorithm to solve the optimization problem.

4.3.1 Cost Function $C(B_m^*, B_n^*)$

The cost function $C(B_m^*, B_n^*)$ is introduced to express the difference between the two functions B_m^* and B_n^*. While the function B_m^* should approximate B_n^* as tightly as possible, it is not clear that the best cost function C is a simple or obvious choice such as the absolute distance between B_m^* and B_n^*. For example, since the burstiness of VBR video limits the number of admitted connections at small delay bounds, it is important that the function B_m^* approximates B_n^* closely for small values of t.

We have evaluated a number of candidate cost functions of the following general form:

$$C(B_m^*, B_n^*) = \int_0^{T_0} \frac{(B_m^*(t) - B_n^*(t))^\alpha}{(t+1)^\beta B_n^*(t)^\gamma}\, dt, \tag{4.9}$$

where T_0 and the exponents α, β, and γ in equation (4.9) determine the shape of the cost function. For example, a selection of $(2,0,0)$ for the (α, β, γ)-tuple results in an approximation where the square of the difference between B_m^* and B_n^* is minimized. However, a least-squares model may not be appropriate since the function B_m^* is required to be larger than B_n^*. We found the following cost function to result in accurate characterizations for the class of small delay bounds ($d \leq 500$ ms):

$$C(B_m^*, B_n^*) = \int_0^{k\,r} \frac{B_m^*(t) - B_n^*(t)}{B_n^*(t)}\, dt \tag{4.10}$$

This cost function measures the amount that B_m^* overestimates the function B_n^* relative to the size of B_n^*.

4.3.2 A Heuristic Algorithm

As we mentioned in Section 4.1, the number of possible (σ, ρ) pairs is infinite, and the selection of a set of pairs that minimizes $C(B_m^*, B_n^*)$ is a combinatorial problem. For this reason, we turn to heuristic approximations for the optimization problem. Here we present a heuristic algorithm that determines m (σ_i, ρ_i) pairs to produce a traffic constraint function B_m^* with low cost $C(B_m^*, B_n^*)$. The algorithm takes as input the function B_n^*, the number of available (σ_i, ρ_i) pairs, the cost function $C(B_m^*, B_n^*)$, and a sensitivity parameter $\epsilon > 0$. The approach of the algorithm is to select initial values for all pairs (σ_i, ρ_i) and then iteratively modify these values to reduce the cost $C(B_m^*, B_n^*)$.

Input: A set of n pairs $\{(\overline{\sigma}_j, \overline{\rho}_j) \mid j = 1, \ldots, n\}$ that define the function B_n^*, the number m of available (σ_i, ρ_i) pairs, a cost function $C(B_m^*, B_n^*)$, and a sensitivity parameter ϵ.

Output: A set of m pairs $\{(\sigma_i, \rho_i) \mid i = 1, \ldots, m\}$ that define the traffic constraint function B_m^*.

1. **Procedure** Parameterize (B_n^*, m, $C(B_m^*, B_n^*)$, ϵ)
2.   **For** $i = 1$ **To** m /* Initialize (σ_i, ρ_i) */
3. $\sigma_i \leftarrow \overline{\sigma}_{\lfloor \frac{in}{m} \rfloor}$
4. $\rho_i \leftarrow \overline{\rho}_{\lfloor \frac{in}{m} \rfloor}$
5.   **End For**
6.   **Do**   /* Greedy modifications */
7.   Cost $\leftarrow C(B_m^*, B_n^*)$
8.   **For** $i = m$ **Down To** 1
9.   Select (σ_i, ρ_i) to minimize $C(B_m^*, B_n^*)$, where $\sigma_{i-1} \leq \sigma_i \leq \sigma_{i+1}$
10.   **End For**
11.   **While** (Cost $- C(B_m^*, B_n^*) > \epsilon$)
12.   **Output** $B_m^* \leftarrow \min_{1 \leq i \leq m} \{\sigma_i + \rho_i t\}$
13. **End Procedure**

Table 4.1: Parameterization algorithm.

The algorithm is presented in Figure 4.1. The initialization of the pairs (σ_i, ρ_i) is shown in steps 2 through 5 of Figure 4.1. Observe that the initial values are a subset of the pairs $\{(\overline{\sigma}_j, \overline{\rho}_j) \mid j = 1, \ldots, n\}$ that determine B_n^*.

The heuristic improves the initial selection by altering the (σ_i, ρ_i) pairs using the iteration shown in steps 6 through 11 of the figure. In each iteration step, the (σ_i, ρ_i) pairs are modified to reduce the cost function C. The iteration terminates when the cost cannot be significantly reduced. The crucial step of the algorithm is step 9, where a single pair (σ_l, ρ_l) is modified to minimize the cost function. During this step, the values of all pairs $\{(\sigma_i, \rho_i) \mid i \neq l\}$ are kept constant, and the pair (σ_l, ρ_l) is selected subject to the constraint that $\sigma_{l-1} < \sigma_l < \sigma_{l+1}$ (with boundary conditions for this selection given by $\sigma_1 \geq 0$

and $\sigma_m \leq \overline{\sigma}_n$). Note that the choice of ρ_l is dependent on σ_l according to the relationship described in equation (4.1).

REMARKS: In the empirical evaluation presented in Section 4.3.3, we select the (σ_i, ρ_i) pair of minimum cost in step 9 through an exhaustive search through all possible values of σ_i. However, with ρ_i expressed in terms of σ_i, it is possible to write $C(B^*_m, B^*_n)$ with σ_i as the only independent variable, and the selection can be determined analytically by setting $\frac{\partial C}{\partial \sigma_i} = 0$. Also, while we do not make guarantees on the running time of the algorithm, the examples that we ran converged rapidly. In all examples using a sensitivity parameter $\epsilon = 0$, no more than six iterations were required.

4.3.3 Empirical Evaluation

We are now ready to evaluate our fast traffic characterization method for VBR video sources by comparing it with other traffic characterization schemes from the literature. With the results from Sections 4.2 and 4.3, our characterization method computes a function B^*_m based on the function $\mathcal{H}R^*_{200} \equiv B^*_n$ which in turn is obtained from the first 200 frames of the empirical envelope E^*. We evaluate the characterization method using the MPEG video traces *Park* and *News* described in Section 4.2.3 and a single FCFS multiplexer at a switch that operates at 155 Mbps.

We compare the traffic characterizations obtained with our method to other schemes that have been considered in the literature. These benchmarks are shown in Figure 4.2, and their parameters are described in the following:

(a) *Peak-rate*: A peak-rate characterization is determined by a single rate parameter ρ_{peak} which is assumed to be the ratio of the size of the largest video frame f_j and the inter-frame time r, i.e., $\rho_{peak} = \dfrac{\max_{0<j\leq N} f_j}{r}$.

(b) *Dual bucket*: In addition to ρ_{peak} described above, the dual bucket scheme employs a pair $(\sigma_{avg}, \rho_{avg})$ where ρ_{avg} is the average traffic rate over the length of the video

Scheme	Parameters	Traffic Constraint Function A^*
Peak-rate	ρ_{peak}	$A^*_{peak}(t) = \rho_{peak} t$
Dual bucket	ρ_{peak}, $(\sigma_{avg}, \rho_{avg})$	$B^*_{db}(t) = \min\{\rho_{peak}\, t\ ,\ \sigma_{avg} + \rho_{avg}\ t\}$
Fixed burst	ρ_{peak}, $(\sigma_{fixed}, \rho_{fixed})$	$B^*_{fixed}(t) = \min\{\rho_{peak}\, t\ ,\ \sigma_{fixed} + \rho_{fixed}\ t\}$
Concave hull	$\{(\hat{\sigma}_j, \hat{\rho}_j) \mid j = 1, \ldots, m\}$	$B^*_{hull}(t) = \min_{1 \le j \le m}\{\hat{\sigma}_j + \hat{\rho}_j\ t\}$
Product	ρ_{peak}, $(\sigma_{product}, \rho_{product})$	$B^*_{product}(t) = \min\{\rho_{peak}\, t\ ,\ \sigma_{product} + \rho_{product}\ t\}$
Distance	ρ_{peak}, $(\sigma_{distance}, \rho_{distance})$	$B^*_{distance}(t) = \min\{\rho_{peak}\, t\ ,\ \sigma_{distance} + \rho_{distance}\ t\}$

Table 4.2: Traffic parameterization schemes with their parameters and traffic constraint functions.

sequence, i.e., $\rho_{avg} = \dfrac{\sum_{j=1}^{N} f_j}{N r}$. The value of σ_{avg} is dependent on ρ_{avg} according to the relationship in equation (4.1).

(c) *Fixed burst*: The scheme outlined in [80] uses a single pair $(\sigma_{fixed}, \rho_{fixed})$ with the burst parameter σ_{fixed} set equal to a "reasonable" buffer size suggested to be either 1000 or 2000 cells, where the parameter ρ_{fixed} is obtained from σ_{fixed} using equation (4.1). We set $\sigma_{fixed} = 1000$ cells since this choice yields better empirical performance. We also add a cell-spacer to enforce the peak rate ρ_{peak} of the connection.

(d) *Concave hull*: The concave hull approach in [103] selects m $(\hat{\sigma}, \hat{\rho})$ pairs for traffic characterization that are taken directly from the concave hull of the empirical envelope $\mathcal{H}E^*$. Consider the n pairs $\{(\hat{\sigma}_j, \hat{\rho}_j) \mid j = 1, \ldots, n\}$ of $\mathcal{H}E^*$, where $\hat{\sigma}_i < \hat{\sigma}_j$ for $i < j$. The parameters selected by the concave hull approach are the m pairs from $\mathcal{H}E^*$ that have the smallest bursts, that is, the pairs $\{(\hat{\sigma}_j, \hat{\rho}_j) \mid j = 1, \ldots, m\}$.

(e) *Product*: In [46] a scheme is proposed that uses the peak rate ρ_{peak} and a pair $(\sigma_{product}, \rho_{product})$, where $\sigma_{product}$ and $\rho_{product}$ are chosen from the candidate set of leaky buckets determined by equation (4.1) such that the product $\sigma_{product} \cdot \rho_{product}$ is minimized.

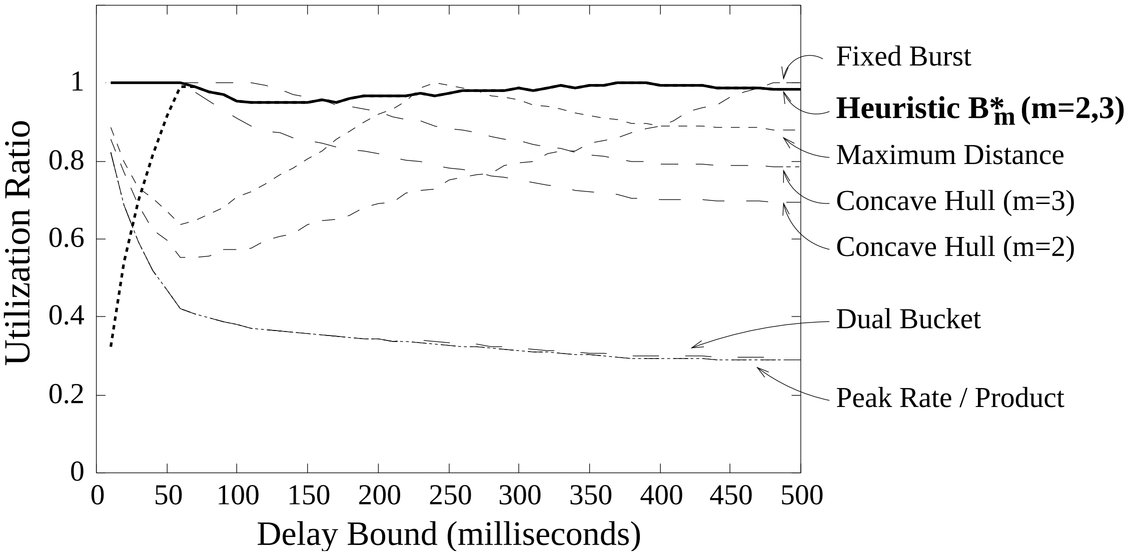
(f) *Distance*: This scheme from [46] uses the peak rate ρ_{peak} and a pair $(\sigma_{distance}, \rho_{distance})$ where $\rho_{distance}$ is selected such that $\delta = \sup_t \{ \frac{\sigma_{distance} + \rho_{distance} t}{t} - \frac{M^*_\epsilon(t)}{t} \}$ is minimized, $M_\epsilon(t) = \inf\{n,\ Pr\{N_t \geq n\} \leq \epsilon\}$ as discussed in Section 4.1, and $\epsilon = 0$ since we seek a worst-case bound.

We evaluate the accuracy of an arbitrary traffic constraint function A^* as follows. We assume that all traffic has the same traffic characterization A^* and identical delay bounds, and we compute the maximum number of admissible connections for all delay bounds as before. Since we wish to evaluate the ability of a particular traffic constraint function to approximate the empirical envelope, we plot the ratio of the number of admissible connections using A^* to the number obtained using the empirical envelope E^*, all as a function of the delay bound. In particular, for a given function A^* we plot:

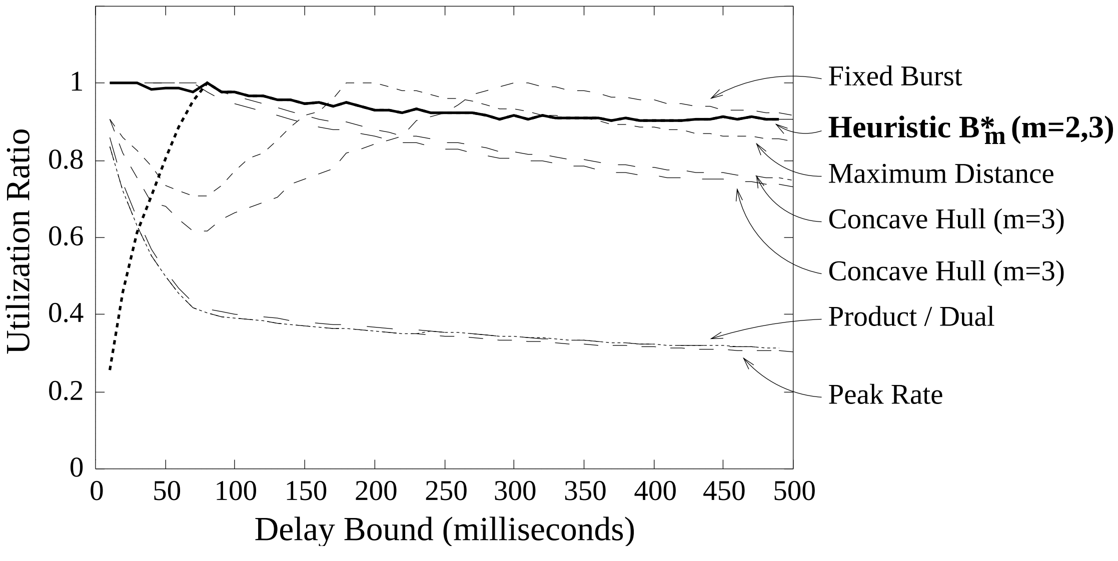
$$\text{Utilization Ratio}(A^*, d) = \frac{\#\text{ admissible connections with } A^* \text{ at delay bound } d}{\#\text{ admissible connections with } E^* \text{ at delay bound } d} \tag{4.11}$$

Since all characterizations A^* considered will necessarily admit fewer connections than the empirical envelope, the metric allows us to determine how closely a particular characterization approximates the empirical envelope. For example, a traffic characterization A^* that admits the same number of connections as the empirical envelope would result in a constant curve Utilization Ratio$(A^*, d) = 1$.

Figures 4.6(a) and 4.6(b) show the utilization ratios of *Park* and *News* connections, respectively, for the entire suite of traffic characterizations described previously, namely, B^*_m, A^*_{peak}, B^*_{db}, B^*_{fixed}, B^*_{hull}, $B^*_{product}$, and $B^*_{distance}$. We depict characterizations B^*_m and B^*_{hull} for both $m = 2$ and $m = 3$ (σ, ρ) pairs.



(a) *Park*



(b) *News*

Figure 4.6: Evaluation of characterization schemes.

The results for our heuristic characterization B_m^* are shown in Figure 4.6 as thick dashed and solid lines for $m = 2$ and and $m = 3$ (σ, ρ) pairs, respectively. For two (σ, ρ) pairs, note that our heuristic achieves a poor utilization for small delay bounds, while it is superior to other characterizations for most delay bounds greater than 50ms. This poor utilization at smaller delay bounds is due to the fact that our heuristic does not select the (σ, ρ) pair with $\rho = \rho_{peak}$. With three (σ, ρ) pairs, our heuristic B_3^* achieves a utilization ratio of over 95% and 91% for all delay bounds in the *Park* and *News* sequences, respectively. The characterization B_3^* produced by our heuristic method that employs three pairs is clearly the best characterization under consideration.

Notice the poor performance of the three characterizations A_{peak}^*, B_{db}^*, and $B_{product}^*$ in both graphs. While a peak-rate characterization yields relatively high utilizations for small delay bounds, the function A_{peak}^* achieves a utilization ratio of less than 40% for delay bounds greater than 60 ms for these video sequences. The additional leaky bucket employed in B_{db}^* and $B_{product}^*$ does not yield significant utilization gains. These three traffic characterizations are notably inferior to the other schemes.

4.4 Case Study: VBR Service with Deterministic Renegotiation

In this section we present a case study that applies our fast traffic characterization method to networks that renegotiate traffic parameters. In a network that employs renegotiation, traffic characterizations are occasionally modified to exploit long-term traffic variations of the VBR video traffic source, possibly leading to increased network utilization [19, 44, 114]. Since a renegotiation scheme requires multiple traffic characterizations for a single connection, a fast traffic characterization scheme such as the one described in this chapter can be used to renegotiate traffic parameters. Here, we first discuss existing renegotiation strategies and point out modifications necessary to use renegotiation with a bounded-delay service. We next show how to apply our traffic characterization scheme to networks that employ renegotiation in a deterministic setting.

4.4.1 Renegotiation of Traffic Characterizations

Dynamic resource allocation schemes are motivated by studies showing that correlations of VBR video traffic occur over long time scales due to the extended duration of scenes [37, 65, 69]. By renegotiating the traffic characterization, for example, after each scene change, one can more accurately specify the traffic on a connection, resulting in a tighter characterization and hence higher network utilization.

Most renegotiation schemes that have been proposed attempt to renegotiate the traffic characterization of a connection whenever its long-term rate changes significantly [19, 44, 114]. Chong et. al. address the problem of predicting the rate changes of a live video source [19]. They consider both a recursive least-square method and an artificial neural network approach for the prediction. In [44], Grossglauer et. al. propose a *Renegotiated Constant Bit Rate* (RCBR) scheme for both stored and live video which adds renegotiation and buffer monitoring to a static CBR service. They present algorithms for partitioning a video sequence into segments based on cost functions for both bandwidth allocation and number of renegotiations. Zhang and Knightly study a renegotiated VBR service for both stored and live video in [114]. Their algorithm for stored video proceeds by identifying the worst-case segment of the video sequence, characterizing this worst-case segment, and then iteratively repeating the procedure on the remaining video sequence after this worst-case segment is removed.

Although the above renegotiation schemes were shown to increase network utilization significantly, they cannot be used in a bounded-delay service. Since these schemes partition a video sequence into a number of segments and calculate a traffic characterization independently for each segment, it is possible that a situation occurs where several connections need to increase their resource allocation even if sufficient resources are not available. In such a scenario, the renegotiation requests cannot be accommodated, and either the video quality or the QoS must be compromised, resulting in a violation of the worst-case guarantees in a bounded-delay service. In the remainder of this section, we present a renegotiation scheme that does not incur the risk of compromising QoS guarantees, Note that this is the first renegotiation scheme proposed so far that is applicable to connections with a worst-case QoS. We use the discussion to demonstrate the effectiveness of our characterization method in such a renegotiation scheme.

4.4.2 Deterministic Renegotiation

A key requirement for a renegotiation scheme in a bounded-delay service is to ensure that the traffic characterization for a connection does not increase, i.e., connections only release resources and do not request additional resources. If the traffic characterizations do not increase, then all renegotiation requests can be satisfied and deterministic QoS guarantees are maintained.

Let the traffic on a video connection be given by A. We assume that the traffic characterization is negotiated at $u+1$ distinct times $\tau_0, \tau_1, \ldots, \tau_u$, where $\tau_i < \tau_j$ if $i < j$ and that the traffic characterization negotiated at time τ_i is given by $A^*_{\tau_i}$. Now, any traffic characterization $A^*_{\tau_i}$ must provide a bound on the worst-case traffic for the remainder of the video sequence, that is, for all i:

$$A^*_{\tau_i}(t) \geq A[\tau_i + \tau, \tau_i + \tau + t] \qquad \forall \tau, t \geq 0 \tag{4.12}$$

Further, to ensure that all renegotiation requests are satisfied, a newly-computed traffic characterization may not request additional resources, that is, we enforce that for all $\tau_i < \tau_j$:

$$A^*_{\tau_i}(t) \geq A^*_{\tau_j}(t) \qquad \text{for all } t \geq 0 \tag{4.13}$$

The condition in equation (4.12) ensures that any function $A^*_{\tau_i}$ is a viable traffic constraint function, while equation (4.13) guarantees that the renegotiation requests can be satisfied.

To show that a set of traffic characterizations $\{A^*_{\tau_i}\}$ can be used in a renegotiation scheme with a bounded-delay service, it is sufficient to show that equations (4.12) and (4.13) are satisfied.

We construct a class of traffic constraint functions $\{E^*_{\tau_i}\}$ that satisfies both equations (4.12) and (4.13) by defining the function $E^*_{\tau_i}$ to be the empirical envelope of the sequence A for all times $t \geq \tau_i$, that is:

$$E^*_{\tau_i}(t) = \sup_{\tau \geq 0} A[\tau_i + \tau, \tau_i + \tau + t] \qquad \forall t \geq \tau_i \tag{4.14}$$

$E^*_{\tau_i}$ is the tightest characterization for the video sequence A for $t \geq \tau_i$. $E^*_{\tau_i}$ satisfies equation (4.12) by definition. To show that equation (4.13) is also satisfied, we note that for two traffic constraint functions $E^*_{\tau_i}$ and $E^*_{\tau_j}$ of the same video sequence A with $\tau_i < \tau_j$, the following holds:

$$E^*_{\tau_j}(t) = \sup_{\tau \geq 0} A[\tau_j + \tau, \tau_j + \tau + t] \leq \sup_{\tau \geq 0} A[\tau_i + \tau, \tau_i + \tau + t] \leq E^*_{\tau_i}(t), \tag{4.15}$$

We have shown that $\{E^*_{\tau_i}\}$ are a class of valid traffic constraint functions that can be used in a bounded-delay service with renegotiation. If a renegotiation occurs τ_i time units into a video sequence, the resource allocation can be calculated according to $E^*_{\tau_i}$. However, the functions $E^*_{\tau_i}$ are similar to the empirical envelope E^* in that they employ a large number of parameters that are expensive to compute. In the next section we show how to apply our fast traffic characterization method to approximate these functions $\{E^*_{\tau_i}\}$.

4.4.3 Application of the Fast Video Characterization Method

Recall that the characterization method presented in Sections 4.2 and 4.3 proceeds in two steps, and its application to $\{E^*_{\tau_i}\}$ is shown in Figure 4.7. In the first step we calculate R^*_{k,τ_i}, the repetition extrapolation of the first k parameters of $E^*_{\tau_i}$, where R^*_{k,τ_i} has the same form as R^*_k given in equation (4.6). However, since R^*_{k,τ_i} is not a viable traffic constraint function, we calculate a $(\vec{\sigma}, \vec{\rho})$-model traffic characterization $\mathcal{H}R^*_{k,\tau_i}$ by computing the concave hull of R^*_{k,τ_i}. In the second step, to reduce the number of (σ, ρ) pairs required, we apply the heuristic algorithm from Section 4.3 to $\mathcal{H}R^*_{k,\tau_i} \equiv B^*_{\tau_i,n}$, yielding $B^*_{\tau_i,m}$.

$$\{E^*_{\tau_i}(jr) \mid j = 1, \ldots, k\} \xrightarrow{\text{Repetition Extension}} R^*_{\tau_i,k} \xrightarrow{\text{Concave Hull}} \mathcal{H}R^*_{\tau_i,k} = B^*_{\tau_i,n} \xrightarrow{\text{Heuristic Algorithm}} B^*_{\tau_i,m}$$

Figure 4.7: Overview of traffic characterization method.

We first consider the class of functions $\{\mathcal{H}R^*_{k,\tau_i}\}$. To show that $\{\mathcal{H}R^*_{k,\tau_i}\}$ can be used in renegotiation, we require that both equations (4.12) and (4.13) are satisfied. Equation (4.12) is satisfied by construction. To show that equation (4.13) is satisfied, we consider two functions R^*_{k,τ_i} and R^*_{k,τ_j}, where $\tau_i < \tau_j$. From equation (4.15), we obtain directly that $R^*_{k,\tau_i}(t) \geq R^*_{k,\tau_j}(t)$ for all t. We can then conclude that $\mathcal{H}R^*_{k,\tau_i}(t) \geq \mathcal{H}R^*_{k,\tau_j}(t)$ for all t, and thus $\{\mathcal{H}R^*_{k,\tau_i}\}$ satisfies equation (4.13).

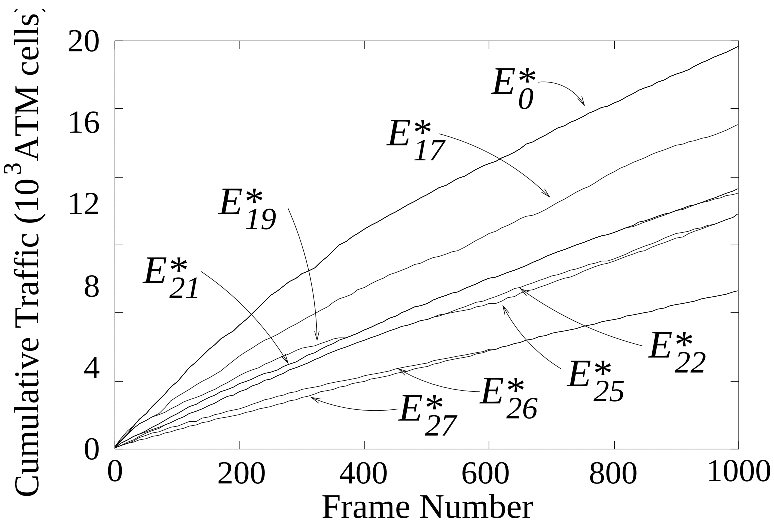
Although the class of functions $\{\mathcal{H}R^*_{k,\tau_i}\}$ are appropriate for use in deterministic renegotiation, we cannot make the same claim about the functions $\{B^*_{\tau_i,m}\}$. Since the heuristic algorithm as presented determines a function $B^*_{\tau_i,m}$ based only on $\mathcal{H}R^*_{k,\tau_i}$, independent of the previous approximation $B^*_{\tau_{i-1},m}$, it is possible for the algorithm to select an approximation $B^*_{\tau_i,m}$ that is larger than $B^*_{\tau_{i-1},m}$ for some values of t. Thus, the condition in equation (4.13) does not necessarily hold. To apply our characterization method to deterministic renegotiation requires a modification of the heuristic where either (1) renegotiation attempts are suppressed at times τ_i whenever $B^*_{\tau_{i-1},m}(t) < B^*_{\tau_i,m}(t)$ for some t, or (2) the search space of the heuristic is modified so that the only pairs (σ, ρ) considered are those that will yield $B^*_{\tau_{i-1},m}(t) \geq B^*_{\tau_i,m}(t)$ for all t.

4.4.4 Empirical Examples

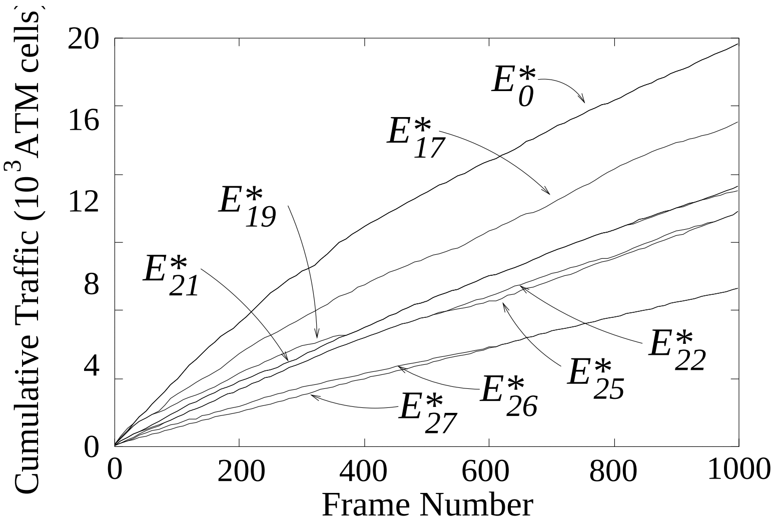
We present examples based on MPEG video sequences to demonstrate the impact of renegotiation on network utilization. For the evaluation, we again use the *Park* and *News* traces described earlier in the chapter.

In the first example we show how the traffic characterization changes as it is renegotiated throughout transmission of the sequence. We consider the class of characterizations $\{E^*_{\tau_i}\}$, where $\tau_i = i$ minutes for $i = 0, \ldots T-1$ and T is the length of the movie in minutes. Since the *Park* and *News* sequences are 28 and 25 minutes long, respectively, we consider 28 different traffic characterizations for *Park* and 25 for *News*. We plot these traffic characterizations in Figure 4.8, where we write $E^*_{\tau_i} = E^*_i$ since $\tau_i = i$. In the figure, we only depict the traffic characterizations E^*_i that are visibly smaller than all traffic characterizations E^*_j with $j < i$. For example, we see in Figure 4.8(a) that all traffic characterizations E^*_j with $0 < j < 17$ appear identical to E^*_0 when plotted.

In the next experiment, we illustrate the average network utilization gain with a deterministic renegotiation scheme using our traffic characterization method. Similar to the experiments in previous sections, we assume that a number of video connections are transmitted on a single 155 Mbps FCFS multiplexer, and we assume that all traffic has the same delay bound d and is of a single traffic type, namely either *Park* or *News*. To evaluate the average performance gain, we assume that the connections are at different transmission points of the stream, resulting in different traffic characterizations due to renegotiation. In particular, we call the frame that is transmitted by a connection at time t the *current frame* at time t, and we assume that the current frames for all connections are uniformly distributed over the entire set of frames $1, \ldots, N$. We consider a scenario in which renegotiation occurs periodically at multiples of a *renegotiation period*. For example, if the renegotiation period is 100 frames, then a connection with current frame 213 uses the traffic characterization computed 200 frames into the sequence based on only frames $200, \ldots, N$.



(a) *Park*



(b) *News*

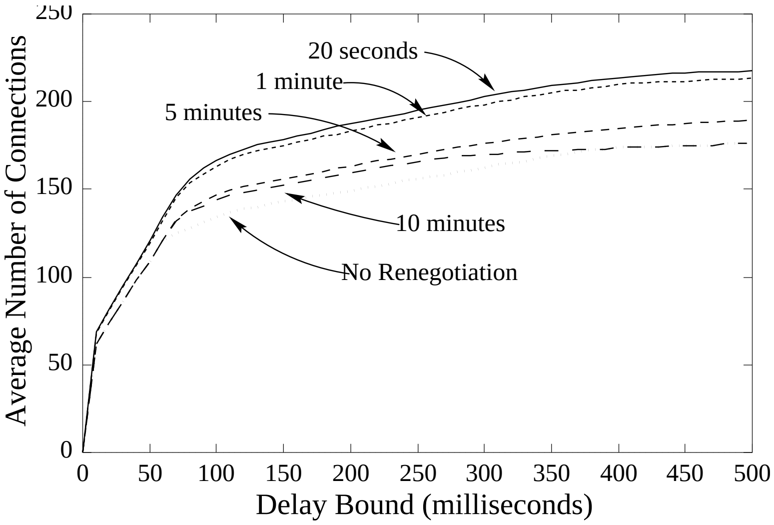
Figure 4.8: Traffic constraint functions $E^*_{\tau_i}$.

The experiment is as follows. Starting with an empty multiplexer, we add connections to the multiplexer, selecting a random current frame for each connection that (together with the renegotiation period) determines its traffic characterization. We continue adding connections as long as the delay bound tests are satisfied, that is, as long as all connections are guaranteed a worst-case delay bound of d. We record the maximum number of admissible connections. This process is repeated 1000 times for each delay bound, and we plot the average number of admissible connections as a function of the delay bound. We obtained similar results for 10 runs of the above experiment.

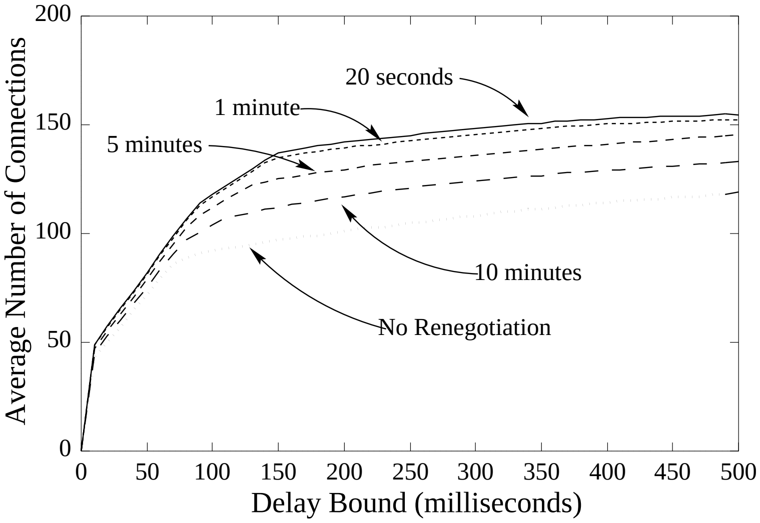
Figures 4.9 and 4.10 depict the number of admissible connections for both the *Park* and *News* sequences for several renegotiation periods. We plot the maximum number of admissible connections as a function of delay bound. Figure 4.9 shows results obtained using $\mathcal{H}R^*_{200,\tau_i}$ for the traffic characterization, while Figure 4.10 uses the functions B^*_{m,τ_i} with two (σ, ρ) pairs. In all graphs, the dotted curves show the utilization obtained when the characterization $\mathcal{H}R^*_{200}$ is employed without any renegotiation. We plot curves corresponding to renegotiation periods of 20 seconds as well as 1, 5, and 10 minutes. For Figure 4.10, we also show the utilization obtained using B^*_2 without negotiation.

We see in Figures 4.9 and 4.10 that the renegotiation period significantly impacts the number of admissible connections. In Figure 4.9(a), note that the number of admissible *Park* connections increases by 20-30% for delay bounds larger than 50 ms if the renegotiation period is less than 1 minute. For the longer renegotiation periods, i.e., 5 minutes and 10 minutes, renegotiation provides gains of about 10%. For the *News* sequence, we see in Figure 4.9(b) that even infrequent renegotiation results in considerable utilization gains.

The plots in Figure 4.10(a) and (b) demonstrate the effectiveness of the heuristic in approximating the functions $\mathcal{H}R^*_{200,\tau_i}$ with B^*_{2,τ_i}. However, since the class of functions B^*_{m,τ_i} are not appropriate for renegotiation without modification, a smaller renegotiation period does not necessarily lead to an increase in network utilization.

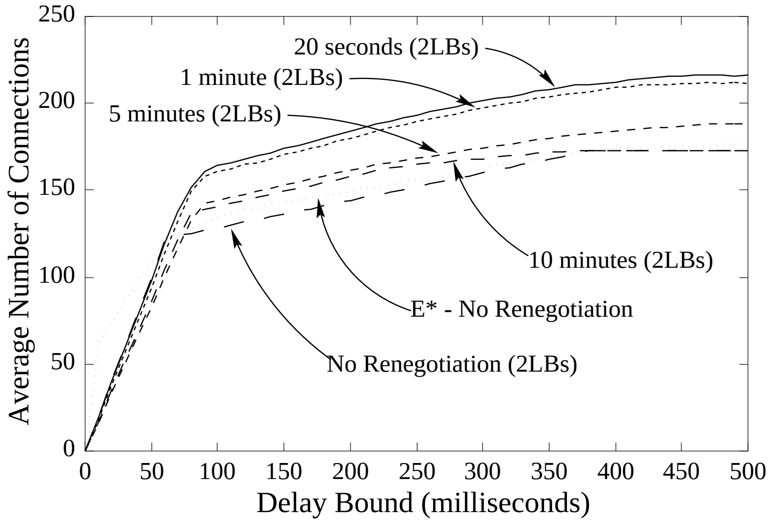


(a) *Park*; $\mathcal{HR}^*_{200,\tau_i}$

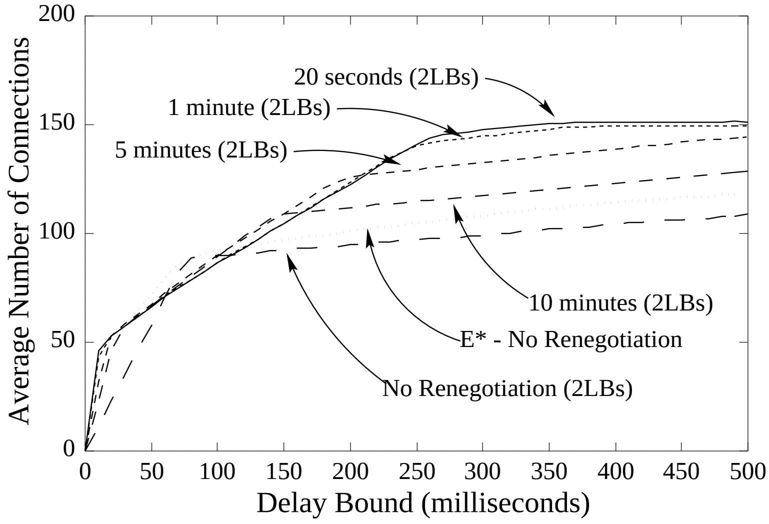


(b) *News*; $\mathcal{HR}^*_{200,\tau_i}$

Figure 4.9: Utilization comparison of $\mathcal{HR}^*_{200,\tau_i}$ for different renegotiation periods τ_i.



(a) *Park*; B^*_{2,τ_i}



(b) *News*; B^*_{2,τ_i}

Figure 4.10: Utilization comparison of B^*_{2,τ_i} for different renegotiation periods τ_i.

4.5 Summary and Remarks

The traffic characterization used for VBR video connections has a significant impact on the number of admissible connections in a network with a bounded-delay service. We presented a method for traffic characterization based on the empirical envelope of a video sequence that uses a two-step process. We first approximate the empirical envelope with a characterization that can be policed by some number of leaky buckets, and we then determine the final characterization that can be policed by a small, fixed number of leaky buckets. Using two MPEG-compressed video sequences, we demonstrated that our characterization method determines accurate characterizations that admit a large number of connections. With the caveat that our experimental evaluation is based on only a small number of video traces, the experiments in this chapter gave the following insights:

- For the MPEG video sequences considered, we saw that as few as 200 parameters of the empirical envelope out of a total of 40,000 are sufficient to yield a characterization that admits the same number of connections as the empirical envelope. This observation suggests that the relevant information of an MPEG sequence is contained in a small segment of the envelope.

- Using our heuristic algorithm, three leaky buckets were shown to be sufficient to admit nearly the same number of connections as the empirical envelope. Based on the good performance of our heuristic algorithm, it may not be worthwhile to investigate more complex algorithms for video characterization.

- The dual leaky bucket scheme was shown to yield poor network utilization in all experiments from Section 4.3.3. However, the poor performance is not due to the fact that only two leaky buckets are used, but rather to a poor selection of leaky bucket parameters. Using a better heuristic algorithm such as the one developed in this chapter, one can achieve markedly higher performance.

- In [103], the numerical examples indicated that a large number of leaky bucket pairs were needed to approximate the empirical envelope. Using our method, the number of leaky buckets needed to achieve performance similar to the envelope is small. This discrepancy can be explained as follows. First, the heuristic algorithm presented here is superior to the concave hull approach from [103]. Second, we note that the examples in this chapter only consider delay bounds up to 500ms, while the examples in [103] consider delay bounds up to 2000ms. For a larger delay bound range, additional leaky buckets are necessary to closely approximate the empirical envelope.

- The deterministic resource renegotiation scheme that we describe is distinguished from previous approaches in that it is appropriate for services that provide constant video quality and deterministic QoS guarantees. The experimental data suggests that the expected utilization gain from this deterministic renegotiation is 20-35%.

5

RPQ$^+$: A Near-Optimal Packet Scheduler for QoS Networks

In addition to the traffic characterization, another important component in the design of such a QoS network is the choice of packet schedulers at network switches that determine the transmission order of packets queued at output buffers. The packet scheduling discipline must be carefully selected so that the network can effectively manage its resources and ensure high achievable network utilization. In this chapter, we consider the design of packet schedulers appropriate for use in networks with a bounded-delay service.

Many packet schedulers have been considered for use in bounded-delay services [31, 32, 41, 52, 111]. The well-known Earliest-Deadline-First (EDF) scheduler has been studied in [31, 38, 39, 74] and is distinguished in that it has optimal *efficiency*: for a given set of connections, EDF can support delay guarantees that are at least as tight as those provided by any other packet scheduler [38, 74]. Each packet arriving to an EDF scheduler is assigned a deadline equal to the sum of its arrival time and associated delay bound, and queued packets are transmitted in increasing order of deadline. Since an EDF scheduler selects packets for transmission according to their deadlines, its implementation requires sorting mechanisms. However, the high overhead costs of sorting prohibit the use of EDF in high-speed networks. For this reason, approximate scheduling disciplines with simpler implementations that achieve an efficiency similar to EDF are needed.

In this chapter we design, analyze, and evaluate a novel packet scheduling method that approximates EDF, the *Rotating-Priority-Queues*$^+$ (RPQ$^+$) scheduler. We will demonstrate that RPQ$^+$ is a near-optimal packet scheduler in the sense that it can approximate the efficiency of the optimal EDF scheduler with arbitrary precision. The RPQ$^+$ scheduler does not require sorting but rather inserts packets into prioritized FIFO queues; the priorities of these FIFO queues are changed (*rotated*) periodically to increase the priority of waiting packets over time. In switches with shared-memory output buffers, the RPQ$^+$ queue rotation can be implemented efficiently through pointer manipulation. The RPQ$^+$ scheduler has the following three key characteristics: (1) The operations of RPQ$^+$ are independent of the number of queued packets. (2) The RPQ$^+$ scheduler can provide worst-case delay guarantees. (3) RPQ$^+$ always yields a higher network utilization than the *Static Priority* (SP) scheduler which does not change the priorities of queues. So far no existing packet scheduler that tries to approximate EDF can satisfy all of the above characteristics (See Section 5.1). We derive necessary and sufficient conditions for schedulability in RPQ$^+$, that is, conditions for which all packets are guaranteed to be transmitted at or before their delay bounds. Using these conditions, we demonstrate that when the rotation period is infinite, i.e., the FIFO queues are never rotated, the efficiency of RPQ$^+$ is identical to SP. We then show that increasing the frequency of queue rotations always yields a higher efficiency, converging to the efficiency of EDF in the limit. We note, however, that greater efficiency requires additional computational overhead in terms of added FIFO queues and more frequent rotations. We compare the efficiency of RPQ$^+$ against other packet schedulers using empirical examples, including an example based on MPEG-compressed video traces [35].

The remainder of this chapter is structured as follows. After discussing related work in Section 5.1, we describe the RPQ$^+$ scheduler in Section 5.2. In Section 5.3 we discuss a shared-memory implementation of RPQ$^+$, and we compare the operational overhead of

RPQ$^+$ with other packet schedulers that approximate the efficiency of EDF. In Section 5.4 we derive necessary and sufficient conditions for schedulability in RPQ$^+$ and show that RPQ$^+$ is a hybrid between SP and EDF. We finally evaluate RPQ$^+$ in Section 5.6 using numerical examples as well as MPEG-compressed video traces.

5.1 Related Work

Recently, several packet schedulers have been considered that approximate EDF with simple implementations [72, 75, 84, 85]. Recall that the main drawback of implementing an EDF scheduler is the sorting operation needed to order packets according to their deadlines. For implementations that use a sorted transmission queue, the complexity of inserting a new packet into the queue is $O(\log N)$, where N is the number of queued packets. At high transmission rates the number of queued packets can be large and the overhead of EDF scheduling can be prohibitive. The approaches in [72, 75, 84, 85] avoid the sorting operation using a similar set of mechanisms. First, all schedulers employ a set of prioritized FIFO queues. Second, each FIFO contains only packets with *laxities* in a certain range, where the laxity of a packet is the time remaining before its deadline. Finally, all schedulers partition the set of connections $\mathcal{C}$ into P connection sets $\{\mathcal{C}_p\}_{1 \leq p \leq P}$ where all connections in $\mathcal{C}_p$ have the identical delay bound d_p.

We first review in Section 5.1.1 the HOL-PJ scheduler presented in [75, 85] that inserts a packet into a FIFO based on its deadline and subsequently moves individual packets to higher-priority FIFOs as dictated by their laxities. In Sections 5.1.2 and 5.1.3 we discuss the priority relabeling architecture [84, 85] and the RPQ scheduler [72], respectively. These packet schedulers are distinct from the first approach in that they do not move individual packets between queues. Instead, they use so-called *calendar queues* [15, 52] that relabel FIFOs periodically to increase the priorities of queued packets.

5.1.1 Head-of-Line with Priority Jumps (HOL-PJ)

Lim and Kobza present in [75] the Head-of-Line with Priority Jumps (HOL-PJ) scheduler. HOL-PJ maintains P FIFO queues labeled FIFO 1, FIFO 2, …, FIFO P, and FIFO q has associated laxity range $[d_{q-1}, d_q]$ with $d_0 = 0$. An arriving packet with delay bound d_q is inserted into FIFO q. To keep packets in the appropriate queues, HOL-PJ maintains a timer for each FIFO queue. The timer for FIFO q expires when the first packet in FIFO q violates the laxity of this queue. Then the packet is dequeued and inserted into FIFO $(q-1)$. A generalization of HOL-PJ called the *recirculation architecture* is presented in [85].

Note that HOL-PJ is an exact implementation of EDF. HOL-PJ has advantages over straightforward implementations of EDF with a single transmission queue in that inserting and removing packets can be performed independent of the number of queued packets. However, HOL-PJ has drawbacks in that it requires a large number of timers and necessitates copying packets between different FIFO queues. Note that the copying of packets can be avoided in shared-memory switches in which FIFO queues are implemented as linked lists.

5.1.2 Priority Relabeling Architecture

In the *priority relabeling architecture* presented by Peha and Tobagi [84, 85], supported delay bounds are of the form $d_p = p\Delta$ for $1 \le p \le P$, where Δ is a parameter of the scheduler. The maximum delay bound supported by the priority relabeling architecture is $P\Delta$. As in HOL-PJ, packets arriving with delay bound d_p are placed into FIFO p. Every Δ time units, the priority relabeling architecture modifies the priorities of the FIFOs by relabeling FIFO p as FIFO $(p-1)$ for all $1 < p \le P$. The laxity range of FIFO p is $[d_{p-1}, d_{p+1}]$ for $1 \le p < P$ and $[d_{P-1}, d_P]$ for FIFO P. Packets that reside in FIFO 1 during such a relabeling are considered as a special case; either (1) all packets in FIFO 1 are dropped, or (2) FIFO 1 and FIFO 2 are concatenated to form the new FIFO 1. Although [85] recommends the former choice, i.e., dropping packets in FIFO 1, for services in which late packets are to be dropped, observe that the scheduler may drop packets that have not violated their deadlines.

As compared to HOL-PJ, the priority relabeling architecture has a much simpler implementation since it requires only a single timer and does not require the movement of queued packets. The relabeling of FIFOs can be accomplished by simply altering an offset in the priority selector [85], and the additional implementation overhead as compared to an SP scheduler is in the relabeling of priorities. To avoid copying packets during the concatenation of FIFO 1 and FIFO 2, the FIFOs must implemented as linked lists in shared memory. Note also that the priority relabeling architecture is not appropriate for bounded-delay services since schedulability conditions are not available and the scheduler may prematurely drop packets in FIFO 1 that have not violated their deadlines.

5.1.3 Rotating-Priority-Queues

The *Rotating-Priority-Queues* (RPQ) scheduler presented in [72] is an approximation of EDF designed to be used with physically separated FIFO buffers that does not require a shared memory. RPQ is similar to the priority relabeling architecture described above in that it supports P delay bounds of the form $d_p = p\Delta$. RPQ maintains $P+1$ FIFO queues with indices $0, 1, \ldots, P$, and every Δ time units the FIFOs are relabeled during a so-called *queue rotation*: FIFO p is relabeled as FIFO $(p-1)$ for $p \geq 1$ and FIFO 0 is relabeled as FIFO P. FIFO 0 is included to hold packets from FIFO 1 that have not violated their deadlines at the time of a queue rotation. Arriving packets are never inserted directly into FIFO 0.

In [74] necessary and sufficient schedulability conditions are derived for RPQ that guarantee the transmission of all packets before their deadlines, and we next state these conditions. Let s_p^{max} denote the maximum transmission time of a packet from $\mathcal{C}_p$. All packets from a set of connections $\mathcal{C}$ will be transmitted prior to their deadlines if and only if the following condition holds for all $t \geq d_1$ [74]:

$$t \geq \sum_{i \in \mathcal{C}_1} A_i^*(t - d_1) + \sum_{p=2}^{P} \sum_{i \in \mathcal{C}_p} A_i^*(t - d_p + \Delta) + \max_{d_q > t - \Delta} s_q^{max} \tag{5.1}$$

A comparison of the condition for RPQ in equation (5.1) with the EDF condition in Theorem 3.1 shows that RPQ can approximate EDF with arbitrary precision if Δ is selected sufficiently small. Note that the schedulability conditions guarantee that FIFO 0 is empty at queue rotations.

Reducing Δ should result in higher efficiency at the expense of higher overhead costs due to more frequent priority relabeling. However, the efficiency of RPQ may be lower than SP for some choices of Δ. In [74] we presented the following pathological example where RPQ cannot admit connections that are admissible by both EDF and SP.

Consider two connection sets 1 and 2 with delay bounds $d_1 = 10$ms and $d_2 = 20$ms and identical traffic constraint functions given as follows ($i = 1, 2$):

$$A_i^*(t) = \left\lfloor \frac{t}{20} \right\rfloor + 1 \tag{5.2}$$

From equation (5.2) we see that the minimum packet interarrival time for any connection is 20 time units.

We now calculate the number of connections of each connection type that can be supported using EDF, SP, and RPQ packet schedulers. We use N_1 and N_2 to denote the number of connections from connection sets 1 and 2, respectively, and we assume in the following that there is some nonzero number of connections in each connection set, i.e., $N_1 > 0$ and $N_2 > 0$.

We first consider EDF scheduling and its schedulability conditions from Theorem 3.1. For all $t \geq d_1$, a set of connections is schedulable using EDF if and only if the following holds:

$$t \geq N_1\, A_1^*(t - d_1) + N_2\, A_2^*(t - d_2) + \max_{d_k > t} s_j^{max} \tag{5.3}$$

Note that it is sufficient to check only times 10 and 20 since (a) it is sufficient to check only times $t \leq 20$ due to construction and (b) 10 and 20 are the only two times $t \leq 20$ for which the right-hand-side of equation (5.3) increases. At times 10 and 20 we obtain $N_1 \leq 9$ and $N_1 + N_2 \leq 20$, respectively.

For SP scheduling, we obtain the following schedulability conditions (see Theorem 3.2):

$$\begin{cases} t \;\geq\; N_1\, A_1^*(t - d_1) + 1 & \text{for all } t \geq d_1 \\ t \;\geq\; N_1\, A_1^*(t) + N_2\, A_2^*(t - d_2) & \text{for all } t \geq d_2 \end{cases} \tag{5.4}$$

From the conditions in equation (5.4), it is easy to see that the restrictions for SP are identical to those of EDF: $N_1 \leq 9$ and $N_1 + N_2 \leq 20$.

For RPQ scheduling, we simplify equation (5.1) to obtain for all times $t \geq d_1$:

$$t \geq N_1\, A_1^*(t - d_1) + N_2\, A_2^*(t + \Delta - d_2) + \max_{d_q > t + \Delta} s_q^{max} \tag{5.5}$$

For the condition in equation (5.5), 10 and $20 - \Delta$ are the two significant times t. For these two values of t, we obtain the restrictions $N_1 \leq 9$ and $N_1 + N_2 \leq 20 - \Delta$. Note that the only finite choice for Δ for which an RPQ scheduler supports the same number of connections as EDF and SP is $\Delta = 0$. In the next section, we present a novel scheduling discipline that addresses the above problem while also retaining many of the desirable properties of the RPQ.

5.2 The Rotating-Priority-Queues$^+$ (RPQ$^+$) Scheduler

In this section we introduce the *Rotating-Priority-Queues*$^+$ (RPQ$^+$) scheduler that approximates the optimal EDF scheduler. Similar to the scheduling disciplines described in the previous section, RPQ$^+$ can be implemented with a set of prioritized FIFO queues that are relabeled periodically. The efficiency of SP provides a lower bound on that of RPQ$^+$, and the efficiency of RPQ$^+$ increases with the frequency of relabeling, approaching that of EDF in the limit. In a shared-memory architecture where FIFO queues are implemented as linked lists, we demonstrate that RPQ$^+$ has low overhead costs that are appropriate for use in high-speed networks. Here we describe the operations of an RPQ$^+$ scheduler and illustrate its operations using a simple example.

5.2.1 RPQ$^+$ Scheduling

Connections submitting traffic to an RPQ$^+$ scheduler are partitioned into P disjoint connection sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P$, and all connections in $\mathcal{C}_p$ have identical delay bounds $d_p = p\Delta$, where Δ is the rotation interval.

The RPQ$^+$ scheduler employs $2P$ ordered FIFO queues, and these FIFOs are indexed as follows, from highest to lowest priority: $0^+, 1, 1^+, 2, 2^+, \ldots, (P-1), (P-1)^+, P$. We refer to the FIFO with index p (p^+) as FIFO p (FIFO p^+). The RPQ$^+$ scheduler always selects a packet from the highest-priority nonempty FIFO for transmission. All packets arriving on a connection in set $\mathcal{C}_p$ are placed in FIFO p. Arriving packets are never placed directly into FIFO p^+ for any p.

Similar to RPQ, the FIFO queues for an RPQ$^+$ scheduler are relabeled every Δ time units. A RPQ$^+$ queue rotation can be viewed as a two-step process: a so-called "concatenation step" and a so-called "promotion step." In the concatenation step, the current FIFO p and FIFO p^+ are merged to form FIFO p for all $1 \leq p < P$. Packets from FIFO p^+ are concatenated to the end of those from FIFO p. In the promotion step, FIFO p is relabeled as the FIFO $(p-1)^+$ for all $1 \leq p \leq P$. Also, a new empty FIFO p is created for all p to hold packet arrivals during the next rotation interval. After the promotion step, all packets reside in some FIFO p^+.

5.2.2 Illustration of RPQ$^+$ Scheduling

The operations of an RPQ$^+$ scheduler are best illustrated by means of a simple example. Figure 5.1 shows an RPQ$^+$ scheduler that supports three connection sets $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ with delay bounds $d_p = p\Delta$ for $p = 1, 2, 3$. Packets from connection set $\mathcal{C}_p$ are labeled p. An RPQ$^+$ scheduler that supports these connection sets requires 6 FIFO queues with indices $\{0^+, 1, 1^+, 2, 2^+, 3\}$. FIFO p^+ is indented for all p to distinguish these queues from FIFO p. In Figure 5.1 packets are assumed to arrive from the left, and packets from connection set $\mathcal{C}_p$ are placed into FIFO p. When a packet is selected for transmission, it is

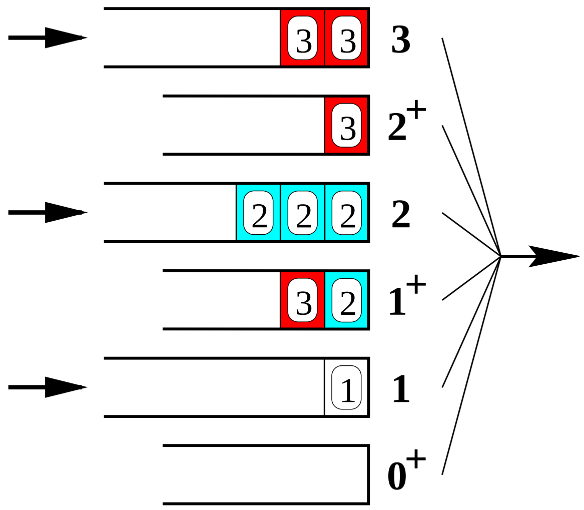
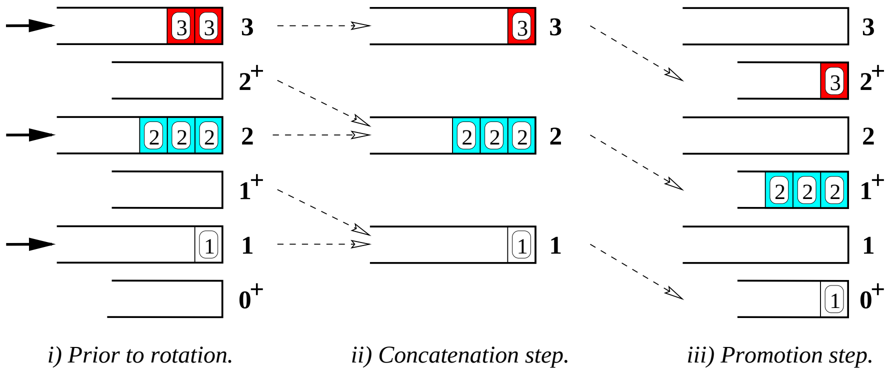


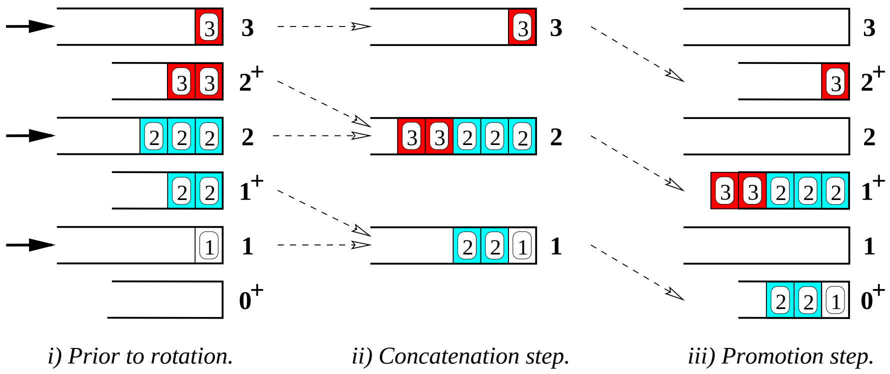
Figure 5.1: RPQ$^+$ scheduler.

assumed to leave the scheduler at right. If packets are in the scheduler as shown in Figure 5.1, the next packet selected will be the packet from connection set $\mathcal{C}_1$ since FIFO 1 is the highest-priority nonempty queue.

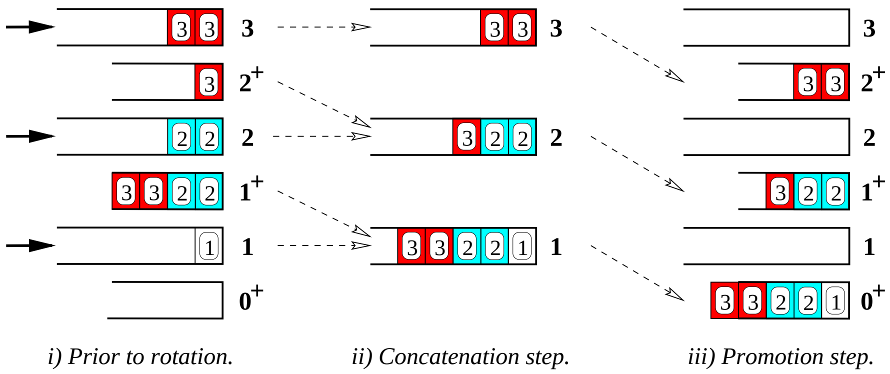
Figure 5.2 illustrates queue rotations and scheduling operations for the RPQ$^+$ scheduler over the course of three rotation intervals. Assuming that the scheduler begins operation at time 0, Figure 5.2(a) shows, from left to right, (i) the state of the queues before the first queue rotation at time Δ, (ii) the concatenation step of the queue rotation, and (iii) the promotion step of the queue rotation. The concatenation step shown in Figure 5.2(a)(ii) involves merging FIFO p and FIFO p^+ into a single FIFO p for all $1 \leq p < P$. We indicate the queues to be merged with dashed lines at the left of the figure. Figure 5.2(a)(iii) shows the promotion of packets from FIFO p to FIFO $(p-1)^+$ for $p = 1, 2, 3$. Note that three new queues FIFO p are included in Figure 5.2(a)(iii) for new arrivals during the next rotation interval.



(a) Rotation at time Δ.



(b) Rotation at time 2Δ.



(c) Rotation at time 3Δ.

Figure 5.2: Example of RPQ$^+$ scheduling operations and queue rotations.

Figure 5.2(b)(i) depicts the state of the queues at time 2Δ. In $[\Delta, 2\Delta)$, packet arrivals from connection set $\mathcal{C}_p$ are placed into FIFO p, but packets from the same connection set that arrived during the previous rotation interval reside in FIFO $(p-1)^+$. The second queue rotation at time 2Δ is illustrated in Figures 5.2(b)(ii) and 5.2(b)(iii). In the concatenation of FIFO p and FIFO p^+ for $1 \leq p < P$, shown in Figure 5.2(b)(ii), all packets from FIFO p^+ are inserted at the tail of FIFO p. Figure 5.2(b)(iii) shows the promotion step of the queue rotation.

Figures 5.2(c)(i) depicts the RPQ$^+$ scheduler at time $3\Delta^-$, and Figures 5.2(c)(ii) and 5.2(c)(iii) illustrate the two phases of the queue rotation at time 3Δ. Note in Figure 5.2(c)(iii) that packets from all 3 connection sets are moved to the highest-priority FIFO 0^+ at time 3Δ.

Observe that we do not specify a location to which packets in FIFO 0^+ are moved during a queue rotation. This problem is not of concern if RPQ$^+$ is used in a bounded delay service where all packets are guaranteed to be transmitted before their deadlines. In this case, the delay bounds for each connection set are selected such that FIFO 0^+ will necessarily be empty at the end of each rotation interval. However, for services other than a bounded-delay service, RPQ$^+$ can be designed to either discard all packets in FIFO 0^+ since they have necessarily violated their deadlines or leave these packets in FIFO 0^+ and concatenate new packets to the end of the FIFO.

5.3 Implementation Issues

Here we investigate the operations required for implementing the RPQ$^+$ queue rotation and demonstrate that RPQ$^+$ is feasible for use in high-speed networks. The overhead for implementing RPQ$^+$ is identical to that of an SP scheduler except for the queue rotations. In switches that use shared-memory output buffering, queue rotations can be implemented with a small number of operations using simple pointer manipulations, meaning that RPQ$^+$ requires little overhead when compared with SP.

In a switch with output buffering, arriving packets are passed through the switching fabric and then buffered until they are selected for transmission on an outgoing link. On a per-port or per-connection basis, the output buffer memory consists of either a single shared memory pool or physically separate memory. Most ATM switches use output buffers, and the majority of these are shared-memory buffers [4, 11, 36]. We assume that RPQ$^+$ operates in switches that have shared-memory output buffers. In such a switch, the FIFO queues of RPQ$^+$ can be implemented using linked lists.

Figure 5.3 shows buffer management for an outgoing link that employs the RPQ$^+$ scheduler. The figure illustrates FIFO queues that are each implemented as a linked list. For each FIFO queue, a "H" and a "T" pointer refer to the head and the tail of the queue, respectively. We use $H(.)$ to denote the head of a queue and $T(.)$ for the tail of the queue; for example, $H(1^+)$ refers to the head of FIFO 1^+. Packets arriving to FIFO p are inserted at $T(p)$, and the scheduler selects packets for transmission from FIFO p at its head $H(p)$.[1]

We now consider the complexity of implementing an RPQ$^+$ queue rotation which we illustrate in Figure 5.4. Figure 5.4(a) shows the state of an RPQ$^+$ scheduler immediately before a queue rotation, while Figures 5.4(b) and 5.4(c) illustrate the concatenation and promotion phases, respectively, of the RPQ$^+$ queue rotation. The operations for the queue rotation involve simple pointer manipulations, and the number of memory accesses in a queue rotation largely determines the total time required. Notice in Figure 5.4(a) that we assume FIFO 0^+ to be empty immediately prior to a queue rotation, an assumption that holds in a bounded-delay service where all packets are transmitted before their deadlines.

Figure 5.4(b) depicts all pointer manipulations required for the concatenation phase of queue rotation. FIFO p and FIFO p^+ are concatenated by linking the tail of FIFO p to the head of the former FIFO p^+. For each concatenation four memory accesses are required: one to obtain the $T(p)$, one to obtain the $H(p^+)$, and two to modify and store the new

[1] Note that $T(p^+)$ is not used since arriving packets are never inserted into FIFO p^+ for any p.

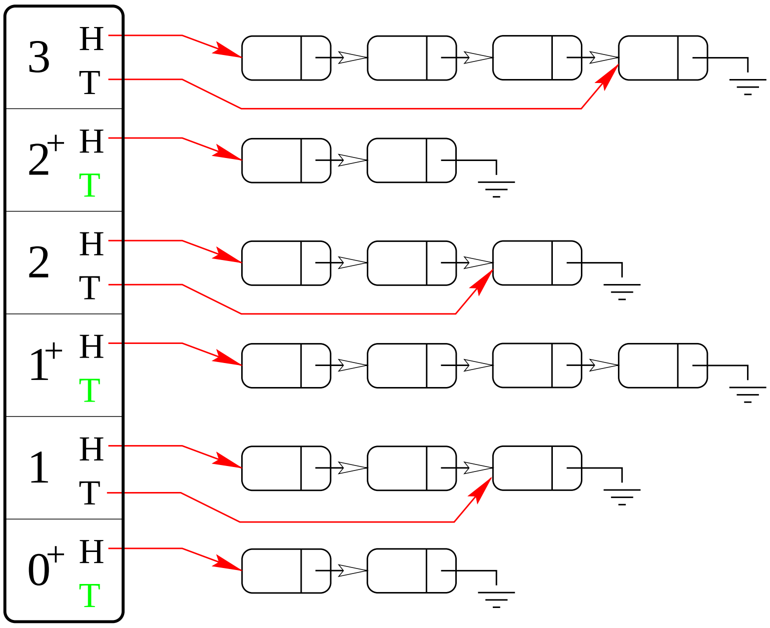
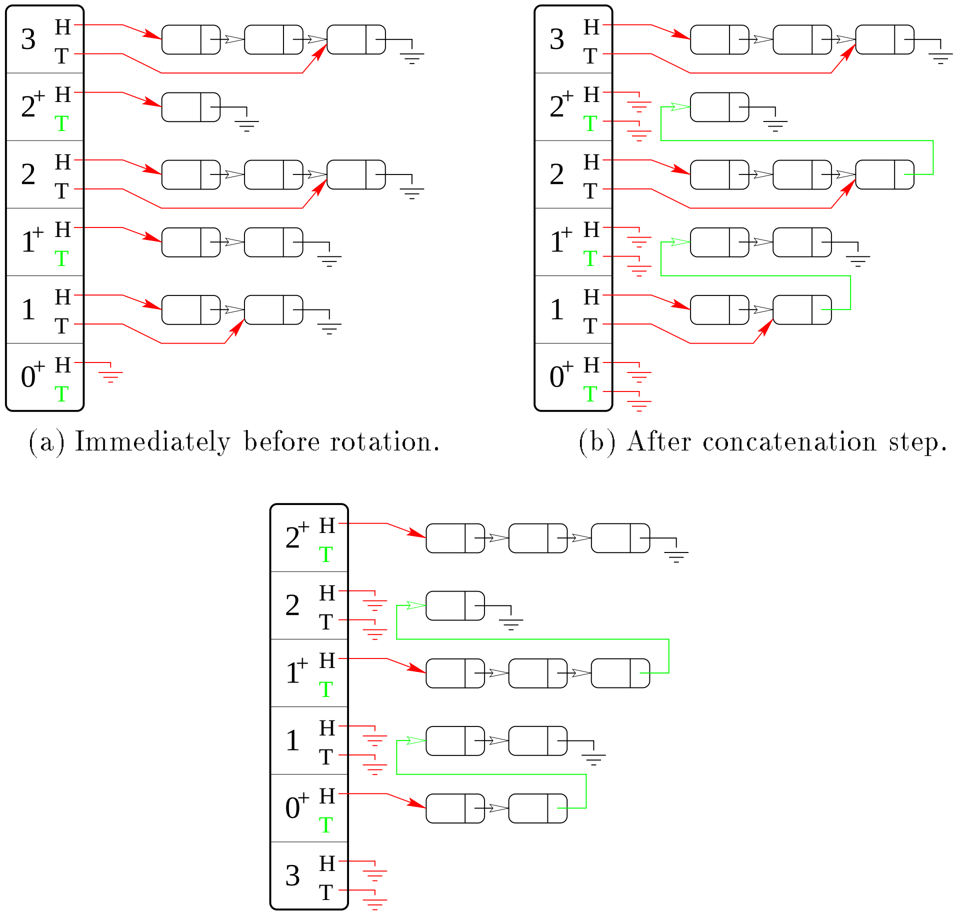


Figure 5.3: Shared-memory output buffer management in RPQ$^+$.

pointer value. Also illustrated in the figure, the pointers $H(p^+)$ and $T(p^+)$ must be cleared, requiring two memory accesses for each of these P FIFOs.

Figure 5.4(c) shows that the promotion phase of RPQ$^+$ can be implemented by simply offsetting priorities. Here FIFO p is relabeled as FIFO $(p-1)^+$, FIFO p^+ is relabeled as FIFO p, and FIFO 0^+ is relabeled as FIFO P. This phase of the queue rotation is implemented exactly the same as the rotation in RPQ and the priority relabeling architecture. For this reason, the pointer manipulations illustrated in Figure 5.4(b) comprise the additional overhead of RPQ$^+$ as compared to RPQ.



(a) Immediately before rotation.

(b) After concatenation step.

(c) After promotion step.

Figure 5.4: Implementation of RPQ^{+} queue rotation.

5.4 RPQ$^+$ Schedulability Conditions

In this section we present schedulability conditions for the RPQ$^+$ scheduler and show that its efficiency is always superior to that of SP. We first derive an expression for the workload transmitted before an arbitrary packet in RPQ$^+$ for the general class of traffic constraint functions described in Chapter 2.1. This expression is central to proving the schedulability conditions and also enables an intuitive understanding of the schedulability conditions. We then present in Theorem 5.1 the exact, that is, necessary and sufficient, schedulability conditions for a general class of traffic constraint functions. We finally show explicitly that RPQ$^+$ is superior to SP and that its efficiency increases monotonically for a certain class of rotation intervals.

5.4.1 Workload Transmitted before an Arbitrary Packet

Assume without loss of generality that a packet from connection j in connection set $\mathcal{C}_p$ arrives to an RPQ$^+$ scheduler at time t. We further assume that the packet is fully transmitted by the scheduler at time $t + \delta$. Here we derive an expression $W^{p,t}(t+\tau)$ that represents the total transmission time of all traffic in the scheduler at time $t + \tau$ to be transmitted before the tagged packet.

The tagged packet arriving at time t arrives after a queue rotation that occurred at time $t - \tau_\Delta$, where $0 \le \tau_\Delta < \Delta$. Queue rotations occur every Δ time units, and so we can express queue rotation times in terms of τ_Δ as follows:

$$\{(t - \tau_\Delta) + i\,\Delta \mid i \text{ an integer}\} \tag{5.6}$$

Let us consider an arbitrary connection set $\mathcal{C}_q$ and determine the times for which packet arrivals from connections $j \in \mathcal{C}_q$ have higher priority than the tagged packet. We consider three cases: connections from the same connection set as the tagged packet ($q = p$), connections of higher priority than the tagged packet ($q < p$), and connections of lower priority ($q > p$).

(a) $\mathbf{q} = \mathbf{p}$: Since all packets from connection set $\mathcal{C}_p$ are transmitted in FIFO order, all packets that arrive before time t are transmitted before the tagged packet. The interval corresponding with $\mathcal{C}_p$ is $[0, t]$.

(b) $\mathbf{q} < \mathbf{p}$: For a higher-priority connection set $\mathcal{C}_q$ with $q < p$, the packets transmitted before the tagged packet are those that arrive before the instant the tagged packet is rotated into FIFO $(q-1)^+$. The tagged packet will be moved to FIFO $(p-1)^+$ during the first queue rotation after t, i.e., time $(t - \tau_\Delta) + \Delta$, and it will be moved to FIFO $(p-2)^+$ at time $(t-\tau_\Delta)+2\Delta$. Thus, at time $(t-\tau_\Delta)+2\Delta$ the tagged packet will be moved into a FIFO with higher-priority than FIFO $(p-1)$, the FIFO into which from connection set $\mathcal{C}_{p-1}$ are placed. More generally, at time $(t - \tau_\Delta) + (n+1)\Delta$ the tagged packet will have a higher priority than new arrivals from connection set $\mathcal{C}_{p-n}$ for $n \geq 1$. Taking into account that the packet departs the scheduler at time $t + \delta$, the time interval corresponding with connection set $\mathcal{C}_q$ for $q < p$ is given by $[0, \min\{t + \delta, (t - \tau_\Delta) + (p - q + 1)\Delta\}]$.

(c) $\mathbf{q} > \mathbf{p}$: For lower-priority connection sets $\mathcal{C}_q$ with $q > p$, only packets that have been rotated into some FIFO r^+ with $r < p$ will be transmitted before the tagged packet. Consider for example packets from connection set $\mathcal{C}_{p+1}$ that arrive up until time $(t-\tau_\Delta)-\Delta$. At time $(t-\tau_\Delta)-\Delta$, these packets will be moved from FIFO $(p+1)$ to FIFO p^+, and they will subsequently be moved to FIFO $(p-1)^+$ at time $t - \tau_\Delta$. Consequently, packets arriving in the time interval $[0, (t-\tau_\Delta)-\Delta]$ will be transmitted before the tagged packet. Note that packets from connection set $\mathcal{C}_{p+1}$ arriving after this time interval will reside in a FIFO with lower priority than FIFO p at time t and hence will not be transmitted before the tagged packet. More generally, for connection set $\mathcal{C}_q$ with $q > p$, packets arriving up until time $(t-\tau_\Delta)+(p-q)\Delta$ will be transmitted before the tagged packet, resulting in the interval $[0, (t - \tau_\Delta) + (p - q)\Delta]$.

The intervals shown above describe the traffic transmitted before the tagged packet, but these intervals do not account for the effects of nonpreemption of packets. In particular, consider a scenario where, at some time prior to the arrival of the tagged packet at time t, there are no packets in the scheduler with arrival times included in the intervals described above. Since the RPQ$^+$ scheduler is work-conserving, some packet not included in the intervals may be transmitted before the tagged packet. We next account for such a nonpreemption in order to accurately quantify the traffic to be transmitted before the tagged packet.

We define $t - \hat{\tau}$ to be the last time before t that the RPQ$^+$ scheduler does not contain packets that are to be transmitted before the tagged packet. Note that such a time is guaranteed to exist since the scheduler is assumed to be empty at time 0. If we use $W_i(\tau)$ to denote the workload in the RPQ$^+$ scheduler from connection $i \in \mathcal{C}$ at time τ, then we can write $\hat{\tau}$ directly from the intervals above as follows:

$$\hat{\tau} = \min\{z \mid \sum_{q=1}^{p} \sum_{i \in \mathcal{C}_q} W_i(t-z) + \sum_{q=p+1}^{P} \sum_{i \in \mathcal{C}_q} W_i(\min\{t-z, (t-\tau_\Delta)+d_p-d_q\}) = 0, z \geq 0\} \quad (5.7)$$

By definition of time $t - \hat{\tau}$, the work transmitted by the RPQ$^+$ scheduler during interval $[t - \hat{\tau}, t + \delta]$ is limited to packets with arrival times during the intervals specified above and the remaining transmission time of some other packet in transmission at time $t - \hat{\tau}$, which we denote by $R(t - \hat{\tau})$.

We are now in a position to explicitly write the workload in the scheduler at time $t + \tau$ that will be transmitted before the packet from connection set $\mathcal{C}_p$ with arrival time t is completely transmitted. This workload is denoted by $W^{p,t}(t+\tau)$ and is given as follows for all τ, $0 \leq \tau \leq \delta$:

$$\begin{aligned} W^{p,t}(t+\tau) &= \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, \min\{t+\tau, (t-\tau_\Delta)+(p-q+1)\Delta\}] + \sum_{i \in \mathcal{C}_p} A_i[t-\hat{\tau}, t] + \\ &\quad \sum_{q=p+1}^{P} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, (t-\tau_\Delta)+(p-q)\Delta] + R(t-\hat{\tau}) - (\hat{\tau}+\tau) \end{aligned} \quad (5.8)$$

The first three terms on the right-hand-side of equation (5.8) account for the arrival intervals derived previously, while the term $R(t-\hat{\tau})$ is the remaining transmission time of the packet transmitted at time $t-\hat{\tau}$. Since by choice of $\hat{\tau}$ the packet scheduler is continuously backlogged for the entire interval $[t-\hat{\tau}, t+\tau]$, the final term accounts for the total workload transmitted during the interval.

Similar to SP, s_p^{max} denotes the maximum transmission time for packets on a connection from set $\mathcal{C}_p$, while s^{min} denotes the minimum packet transmission time for any packet. We assume that the transmission time of the tagged packet is given by s, where $s^{min} \leq s \leq s_p^{max}$. Since the tagged packet completes transmission at time $t+\delta$, the packet must begin transmission at time $t+\delta-s$, a time at which the total workload to be transmitted before or with the tagged packet is s, the transmission time of the packet itself. We can determine the departure time $t+\delta$ of the tagged packet using the following equation:

$$\delta = s + \min\{z \mid W^{p,t}(t+z) = s, z \geq 0\} \tag{5.9}$$

5.4.2 RPQ$^+$ Schedulability Conditions and Properties of RPQ$^+$

Here we present the necessary and sufficient conditions for schedulability in an RPQ$^+$ scheduler. These conditions assume that all connections have traffic constraint functions as described in Chapter 2.1.

Theorem 5.1 *A set $\mathcal{C}$ of connections that is given by $\{A_i^*, d_i\}_{i\in\mathcal{C}}$ is RPQ$^+$-schedulable with rotation interval Δ if and only if for all priorities p and for all $t \geq 0$ there exists a τ with $0 \leq \tau \leq d_p - s^{min}$ such that:*

$$t+\tau \geq \sum_{q=1}^{p-1}\sum_{i\in\mathcal{C}_q} A_i^*(\min\{t+\tau, t+d_p-d_q+\Delta\}) + \sum_{q=p}^{P}\sum_{i\in\mathcal{C}_q} A_i^*(t+d_p-d_q) - s^{min} + \max_{r, d_r > t+d_p} s_r^{max} \tag{5.10}$$

Here we include a sketch of the proof of sufficiency of Theorem 5.1. The complete proof of the theorem is given in Section 5.5.1.

Proof Idea for Sufficiency of Theorem 5.1:

Assuming that the inequality in equation (5.10) holds for all times $t \geq 0$, we must show that an arbitrary packet from connection $j \in \mathcal{C}_p$ arriving at time t does not violate its deadline.

Starting with equation (5.8), we provide an upper bound on the workload transmitted before the tagged packet. In the worst case, $\tau_\Delta = 0$, the traffic $A_i[t_1, t_2]$ on connection i is at most $A_i^*(t_2 - t_1)$, and $R(t - \hat{\tau})$ is limited to $\max_{r, d_r > \hat{\tau} + d_p} s_r^{max}$. Combining these observations with equations (5.8) and (5.10), we can show that there exists some τ $(0 \leq \tau \leq d_p - s^{min})$ such that $W^{p,t}(t + \tau) \leq s^{min}$. Since the transmission time of the packet must be at least s^{min}, the packet will complete transmission at or before time $t + d_p$ as required. $\square$

In order to compare RPQ$^+$ with EDF, we next present a sufficient (but not necessary) schedulability condition for RPQ$^+$ that has a formulation similar to the exact EDF conditions in Theorem 3.1. We obtain these conditions directly from Theorem 5.1 by substituting $\tau = d_p - s^{min}$ in equation (5.10).

Corollary 5.1.1 *Given a set $\mathcal{C}$ of connections that is given by $\{A_i^*, d_i\}_{i \in \mathcal{C}}$, the connections are RPQ$^+$-schedulable with rotation interval Δ if for all priorities p and for all $t \geq d_p$ the following condition holds:*

$$t \geq \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(t - d_q + \Delta) + \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(t - d_q) + \max_{r, d_r > t} s_r^{max} \tag{5.11}$$

Comparing the condition in equation (5.11) with the EDF condition, we see that the only difference in the two conditions is the rotation interval Δ. We see that these two conditions become identical in the limit as $\Delta \rightarrow 0$, verifying that an RPQ$^+$ scheduler effectively approximates EDF with arbitrary precision.

The RPQ$^+$ scheduler is designed to be a hybrid between SP and EDF in the sense that (1) RPQ$^+$ always achieves an efficiency at least as good as SP, (2) the efficiency of RPQ$^+$

is nondecreasing as the rotation interval Δ is reduced[2], and (3) the efficiency of RPQ$^+$ approaches that of EDF as $\Delta \to 0$. The latter two claims are easy to show; the second holds since the right-hand-side of equation (5.10) increases with Δ, while the final claim follows from verifying that the RPQ$^+$ condition in equation (5.11) and the EDF condition from Theorem 3.1 are identical for $\Delta = 0$. We conclude this section by arguing that any set of connections schedulable with SP are also schedulable with RPQ$^+$.

Our argument relies on a necessary condition for SP schedulability.

Lemma 5.1 *If a set $\mathcal{C}$ of connections given by $\{A_i^*, d_i\}_{i \in \mathcal{C}}$ is SP-schedulable, then all priorities p and for all $t \geq 0$ there exists a τ with $\tau \leq d_p - s^{min}$ such that:*

$$t + \tau \geq \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(t+\tau) + \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(t + d_p - d_q) - s^{min} + \max_{r, d_r > t + d_p} s_r^{max} \tag{5.12}$$

Here we provide a proof sketch of Lemma 5.1; a complete proof is given in Section 5.5.2.

Proof Idea:

We prove Lemma 5.1 by contradiction. We assume that equation (5.12) is violated for some priority p and time t for all $0 \leq \tau \leq d_p - s^{min}$. We next construct a scenario in which some packet in connection set $\mathcal{C}_u$ $(u \geq p)$ must have a deadline violation at or before time $t + d_p$.

Assume that a packet of maximal size from connection $k \in \mathcal{C}_r$, with $d_r > t + d_p$ arrives to an empty scheduler immediately prior to time 0, and at time 0 all connections $i \in \mathcal{C}_q$ with $d_q \leq t + d_p$ submit traffic according to A_i^*, with the exception that for $\mathcal{C}_q$ $(q \geq p)$ a packet with transmission time s^{min} is delayed until time $t + d_p - d_q$. All of these packets, which we refer to collectively as the *delayed packets*, have deadlines at time $t + d_p$.

The last of the delayed packets transmitted is on some connection $j \in \mathcal{C}_u$ $(u \geq p)$, and we call this packet the *tagged packet*. Note that the tagged packet will necessarily reside in

[2] Recall that Δ must evenly divide all desired delay bounds since the supported delay bounds are of the form $d_p = p\Delta$ for all p.

the scheduler at time t. Assuming that the tagged packet begins transmission at time $t+\delta$, the traffic to be transmitted before this packet at time $t+\tau$, $0 \le \tau \le \delta$ includes: (a) All traffic from lower-priority connection sets $\mathcal{C}_q$ $(q \ge p)$ that arrives up to time $t + d_p - d_q$, i.e., $\sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(t + d_p - d_q)$. Note that all of this traffic has arrived to the scheduler by time t. (b) All traffic from higher-priority connection sets $\mathcal{C}_q$ $(q < p)$ that arrives to the scheduler up until time $t+\tau$, i.e., $\sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(t+\tau)$. (c) Due to nonpreemption, the packet arriving before time 0 with transmission time $\max_{d_r > t + d_p} s_r^{max}$. Combining the above observations with our assumption, we find that the tagged packet from connection set $\mathcal{C}_u$ with transmission time s^{min} will not begin transmission until after time $t + d_p - s^{min}$ and will therefore have a deadline violation. $\square$

With Lemma 5.1, the necessary condition for SP in equation (5.12) implies the sufficient condition for RPQ$^+$ in equation (5.10). Thus RPQ$^+$ always achieves an efficiency at least as high as SP, and so we have shown that RPQ$^+$ is a hybrid of SP and EDF.

5.5 Proof for RPQ$^+$ Schedulability

5.5.1 Proof of Theorem 5.1

We first prove the sufficiency of the conditions in equation (5.10), and following that is the proof of necessity.

(a) Sufficiency of Theorem 5.1

Here we show that the RPQ$^+$ scheduler will transmit all packets before their deadlines if the inequality in equation (5.10) holds for all times $t \ge 0$. We consider without loss of generality a packet from connection $j \in \mathcal{C}_p$ with transmission time s $(s_p^{min} \le s \le s_p^{max})$ that arrives to the scheduler at time t. Such a packet, which we call the *tagged packet*, has a deadline at time $t + d_p$. To show this, it is sufficient to find some time $\overline{\tau}$ with $\overline{\tau} \le t + d_p - s$ such that $W^{p,t}(t + \overline{\tau}) \le s$.

We consider the workload $W^{p,t}$ transmitted before the tagged packet as given in equation (5.8). The workload is maximal when $\tau_\Delta = 0$, and so we obtain the following bound on $W^{p,t}$:

$$\begin{aligned} W^{p,t}(t+\tau) \quad \leq \quad & \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, \min\{t+\tau, t+(p-q+1)\Delta\}] + \\ & \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, t+(p-q)\Delta] + R(t-\hat{\tau}) - (\hat{\tau}+\tau) \end{aligned} \tag{5.13}$$

Note that we were able to rewrite the second and third terms from equation (5.8) as a single term in equation (5.13). We further constrain terms in the workload expression using the following two inequalities based on the definition of the traffic constraint function A_i^*:

$$\begin{aligned} & \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, \min\{t+\tau, t+(p-q+1)\ \}] \quad \leq \\ & \qquad \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\min\{\hat{\tau}+\tau, \hat{\tau}+(p-q+1)\Delta\}) \end{aligned} \tag{5.14}$$

$$\sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i[t-\hat{\tau}, t+(p-q)\Delta] \leq \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{\tau}+(p-q)\Delta) \tag{5.15}$$

Combining the inequalities in equations (5.14), and (5.15) with the workload expression in equation (5.13), we obtain:

$$\begin{aligned} W^{p,t}(t+\tau) \quad \leq \quad & \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\min\{\hat{\tau}+\tau, \hat{\tau}+(p-q+1)\Delta\}) + \\ & \sum_{q=p+1}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{\tau}+(p-q)\Delta) + R(t-\hat{\tau}) - (\hat{\tau}+\tau) \end{aligned} \tag{5.16}$$

We now consider $R(t-\hat{\tau})$, the remaining transmission time of a packet in transmission at time $t-\hat{\tau}$. Such a packet from a connection $k \in \mathcal{C}_r$ has arrival time $t-t_0$, where $t_0 > \hat{\tau}$. By choice of $t-\hat{\tau}$, the time $t-t_0$ is restricted as follows:

$$t - t_0 > (t - \tau_\Delta) + d_p - d_r \tag{5.17}$$

Eliminating t from equation (5.17) and using the fact that $t_0 > \hat{\tau}$, we obtain:

$$d_r > \hat{\tau} + d_p - \tau_\Delta \tag{5.18}$$

Now, the right-hand-side of equation (5.13) is largest if $\tau_\Delta = 0$, and since the maximum transmission time for a packet from a connection in class $\mathcal{C}_r$ is given by s_r^{max}, we find the following restriction on $R(t - \hat{\tau})$:

$$R(t-\hat{\tau}) \leq \max_{r; d_r > \hat{\tau} + d_p} s_r^{max} \tag{5.19}$$

Combining the inequality from equation (5.19) with our workload expression in equation (5.16), we obtain the following:

$$\begin{aligned} W^{p,t}(t+\tau) \quad \leq \quad & \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\min\{\hat{\tau}+\tau, \hat{\tau}+(p-q+1)\Delta\}) + \\ & \sum_{q=p+1}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{\tau}+(p-q)\Delta) + \max_{r; d_r > \hat{\tau}+d_p} s_r^{max} - (\hat{\tau}+\tau) \end{aligned} \tag{5.20}$$

With the condition in equation (5.10) we know that there exists some $\overline{\tau}$ $(0 \leq \overline{\tau} \leq d_p - s_p^{min})$ such that:

$$W^{p,t}(t+\overline{\tau}) \leq s_p^{min} \tag{5.21}$$

Since the transmission time s is such that $s \geq s_p^{min}$, the condition in equation (5.21) implies that the tagged packet is in transmission at time $t + \overline{\tau}$ and will complete transmission at time $t + \overline{\tau} + s_p^{min}$ Since $\overline{\tau} + s_p^{min} \leq d_p$, the tagged packet will be fully transmitted at time $t + d_p$ as required, completing the proof of sufficiency. □

(b) Necessity of Theorem 5.1

Assume that there is a violation of the condition in equation (5.10). That is, for some connection set $\mathcal{C}_p$ and some time $\hat{t} \geq 0$, the following inequality holds for all τ with $0 \leq \tau \leq d_p - s_p^{min}$:

$$\hat{t}+\tau < \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\min\{\hat{t}+\tau, \hat{t}+d_p-d_q+\Delta\}) + \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t}+d_p-d_q) - s_p^{min} + \max_{r, d_r > \hat{t}+d_p} s_r^{max} \tag{5.22}$$

To show necessity of Theorem 5.1, we construct a feasible sequence of arrivals in which some packet will have a deadline violation.

We consider a scenario in which the RPQ$^+$ scheduler is empty up until time 0^-. At time 0^- assume that a packet of maximal size from a connection k in a connection set $\mathcal{C}_r$ where $d_r > \hat{t} + d_p$. Such a packet has transmission time $\max_{r;d_r > \hat{t}+d_p} s_r^{max}$. Further assume that all connections in connection sets $\mathcal{C}_q$ with $d_q \leq \hat{t} + d_p$ submit a maximal amount of traffic starting at time 0, i.e., at time t, a connection i has submitted $A_i^*(t)$. There is one exception to the traffic from these connections: For some connection j with $j \in \mathcal{C}_p$, we delay the arrival of an amount of traffic with transmission time s_p^{min} that would arrive before time $\hat{t}$ such that it arrives at time $\hat{t}$. Formally, if the last packet arrival from connection $j \in \mathcal{C}_p$ before time $\hat{t}$ occurs at time $\hat{t} - z$ where:

$$z = \min\{z' \mid A_j^*(\hat{t} - z') < A_j^*(\hat{t}), z' \geq 0\}, \tag{5.23}$$

then a packet with transmission time s_p^{min} is split off from this packet and delayed until time $\hat{t}$. We call this packet the *"tagged packet"*. Note that such a packet can be constructed if the packet transmission time for all connections $i \in \mathcal{C}_q$ is either constant (with $s_q^{min} = s_q$) or is such that $s_q^{min} \leq s_q^{max}/2$.

We also assume without loss of generality that a queue rotation occurs at time $\hat{t}$, i.e., $\tau_\Delta = 0$. Based on the above scenario, the workload $W^{p,\hat{t}}(\hat{t} + \tau)$ to be transmitted before the tagged packet at time $\hat{t} + \tau$ can be calculated as follows using equation (5.8):

$$\begin{aligned} W^{p,\hat{t}}(\hat{t} + \tau) \quad = \quad & \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\min\{\hat{t} + \tau, \hat{t} + (p - q + 1)\Delta\}) + \\ & \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + (p - q)\Delta) + \max_{r;d_r > \hat{t}+d_p} s_r^{max} - \hat{t} \end{aligned} \tag{5.24}$$

Now, combining equation (5.24) with our assumption in equation (5.22), we find that, for all τ with $0 \leq \tau \leq d_p - s_p^{min}$:

$$W^{p,\hat{t}}(\hat{t} + \tau) > s_p^{min} \tag{5.25}$$

Thus, the tagged packet will not begin transmission before time $\hat{t} + d_p - s_p^{min}$ and will therefore have a deadline violation. $\square$

5.5.2 Proof of Lemma 5.1

To prove necessity of the condition in Lemma 5.1, we assume towards contradiction that equation (5.12) is is violated at some time $\hat{t}$, that is, there exists a priority p such that for all $\tau \leq d_p - s^{min}$:

$$\hat{t} + \tau < \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + \tau) + \sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + d_p - d_q) - s^{min} + \max_{r, d_r > \hat{t} + d_p} s_r^{max} \tag{5.26}$$

We will show that a packet from some connection $j \in \mathcal{C}_u$ with $u \geq p$ at the SP scheduler will have a deadline violation at or before time $t + d_p$.

We assume without loss of generality that arrivals to the SP scheduler are as follows. The scheduler is empty before time 0^-, and at time 0^- a packet arrives to the scheduler from connection $k \in \mathcal{C}_r$, where $d_r > \hat{t} + d_p$ and the packet requires maximal transmission time. The transmission time of such a packet is given by $\max_{r, d_r > \hat{t} + d_p} s_r^{max}$. Beginning at time 0 all connections in connection sets $\mathcal{C}_q$ with $d_q \leq \hat{t} + d_p$ submit a maximal amount of traffic to the scheduler, i.e., the traffic submitted to the network from a connection i at time t is $A_i^*(t)$, with the exception that for all connection sets $\mathcal{C}_q$, $(q \geq p)$, an amount of traffic with transmission time s^{min} arriving before or at time $\hat{t} + d_p - d_q$ is delayed until time $\hat{t} + d_p - d_q$. In this construction there is a packet from each lower-priority connection set $\mathcal{C}_q$ that has a deadline at time $\hat{t} + d_p$ and minimal transmission time. We refer to these packets as the *delayed packets*.

We consider the last of all delayed packets to be transmitted. This packet, which we call the *tagged packet*, is on some connection $j \in \mathcal{C}_u$ with $u \geq p$ and arrived to the scheduler at time $\hat{t} + d_p - d_u$. Note that the tagged packet begins transmission after time $\hat{t}$ since the arrival from connection set $\mathcal{C}_p$ occurs at that time. Assuming that the packet begins transmission at time $\hat{t} + \delta - s^{min}$, the workload $W^{u,\hat{t}+d_p-d_u}(\hat{t} + \tau)$ transmitted before the tagged packet in time interval $[\hat{t}, \hat{t} + \delta - s^{min}]$ by an SP scheduler includes the following:

(a) Since the tagged packet is transmitted after all other delayed packets, workload from connection sets $\mathcal{C}_q$ $(q \geq p)$ arriving up until time $\hat{t} + d_p - d_q$ are served before the tagged packet, i.e., the traffic $\sum_{q=p}^{P} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + d_p - d_q)$.

(b) Given that an SP scheduler always transmits the waiting packet with the highest priority, and also given that $u \geq p$, all traffic from connection sets $\mathcal{C}_q$ $(q < p)$ in the scheduler at time $\hat{t} + \tau$ will be transmitted before the tagged packet. Thus, the traffic $\sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + \tau)$ will be served before the tagged packet.

(c) The low-priority packet arriving at time 0^- with transmission time $\max_{r, d_r > \hat{t}+d_p} s_r^{max}$ is the only packet in the scheduler at time 0^-, and so it will be transmitted before the tagged packet.

Thus, the workload to be transmitted before the tagged packet with deadline $\hat{t} + d_p$ can be bounded as follows:

$$W^{u,\hat{t}+d_p-d_u,\hat{t}}(\hat{t} + \tau) \geq \sum_{q=1}^{p-1} \sum_{i \in \mathcal{C}_q} A_i^*(\hat{t} + \tau) + \sum_{i \in \mathcal{C}_p} A_i^*(\hat{t}) + \max_{r, d_r > \hat{t}+d_p} s_r^{max} - (\hat{t} + \tau) \tag{5.27}$$

Combining equation (5.27) with our assumption in equation (5.26), we find that, for all $0 \leq \tau \leq d_p - s^{min}$, $W^{u,\hat{t}+d_p-d_u,\hat{t}}(\hat{t} + \tau) > s^{min}$, and thus the tagged packet will not be transmitted before its deadline, and a deadline violation will occur. □

5.6 Evaluation

In this section we compare the efficiency of the RPQ$^+$ scheduler against schedulers EDF, SP, and RPQ using empirical examples. Two sets of experiments are included. The first experiment uses traffic policed by leaky buckets, while the second experiment uses traces of MPEG-compressed video. In both experiments we use the most accurate, i.e., necessary and sufficient, admission control mechanisms for each packet scheduler to obtain a precise comparison of the efficiency of the schedulers. We use the schedulability conditions from Theorem 5.1 for RPQ$^+$; the conditions from Theorems 3.1 and 3.2 for EDF and SP, respectively; and the condition in equation (5.1) for RPQ.

5.6.1 Numerical Example

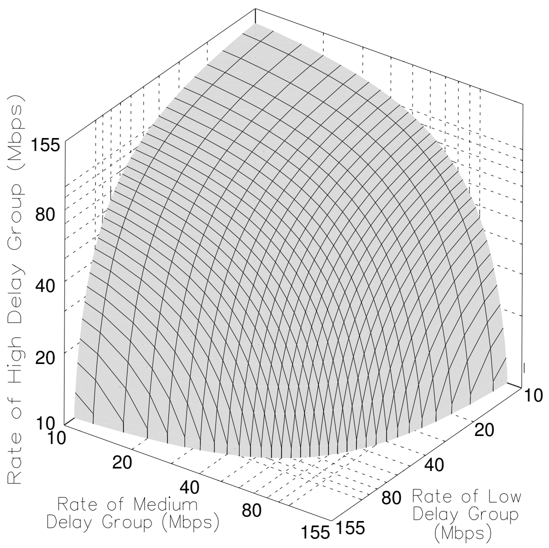
In the first experiment we compute the *schedulable region* of the packet schedulers for a set of three connection groups using an approach similar to one used in [49, 74]. We vary the traffic rate of each connection group and use a surface plot to illustrate the rates for which all delay bounds are guaranteed. We can compare the efficiencies of different packet schedulers graphically by comparing the volumes of their schedulable regions.

We consider connections supported at a single packet scheduler that transmits packets at 155 Mbps, a rate that corresponds to OC-3. The three connection groups have traffic that conforms to the (σ, ρ) traffic model, with traffic constraint functions A^* of the form:

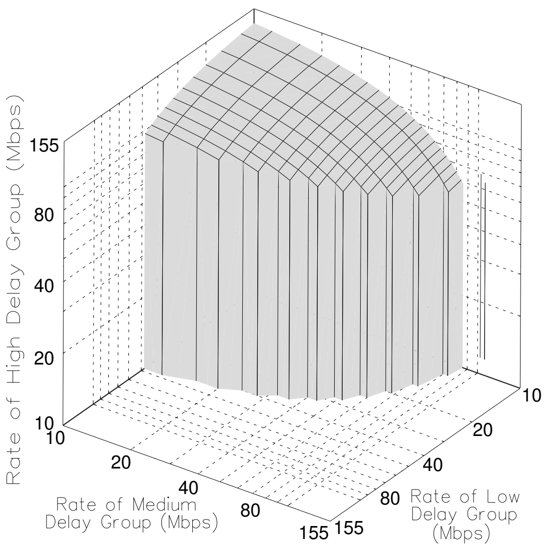
$$A^*(t) = \sigma + \rho t \tag{5.28}$$

Table 5.1 shows the traffic and QoS parameters for all connection groups. For a connection group with index j, the table shows the delay bound d_j at the scheduler as well as the burst σ_j and the range of rates ρ_j. The bursts b_j in Table 5.1 are given in 53-byte ATM cells. In the example, we vary the rate parameter ρ_j between 10 and 155 Mbps.

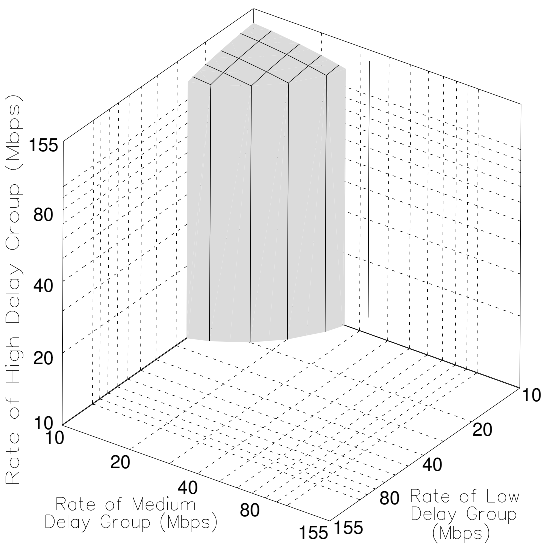
We present our results with three sets of graphs in Figures 5.5-5.8. In the first two figures, we illustrate the schedulable region for different schedulers using three-dimensional surface plots. We show the transmission rates for which all connection groups are admissible; the volume beneath a curve includes all operating points for which all packets are guaranteed to be transmitted before their deadlines. Note that the axes in these figures have a logarithmic scale. The last figure, Figure 5.8, is a two-dimensional plot that summarizes all results in a single graph.



(a) Schedulable Region without Delay Constraints.



(b) EDF Scheduler.



(c) SP Scheduler.

Figure 5.5: Benchmark schedulable regions.

	Index j	Delay Bound d_j	Burst σ_j	Rate ρ_j
Low Delay Group	1	12 ms	4000 cells	10-155 Mbps
Medium Delay Group	2	24 ms	2000 cells	10-155 Mbps
High Delay Group	3	36 ms	4000 cells	10-155 Mbps

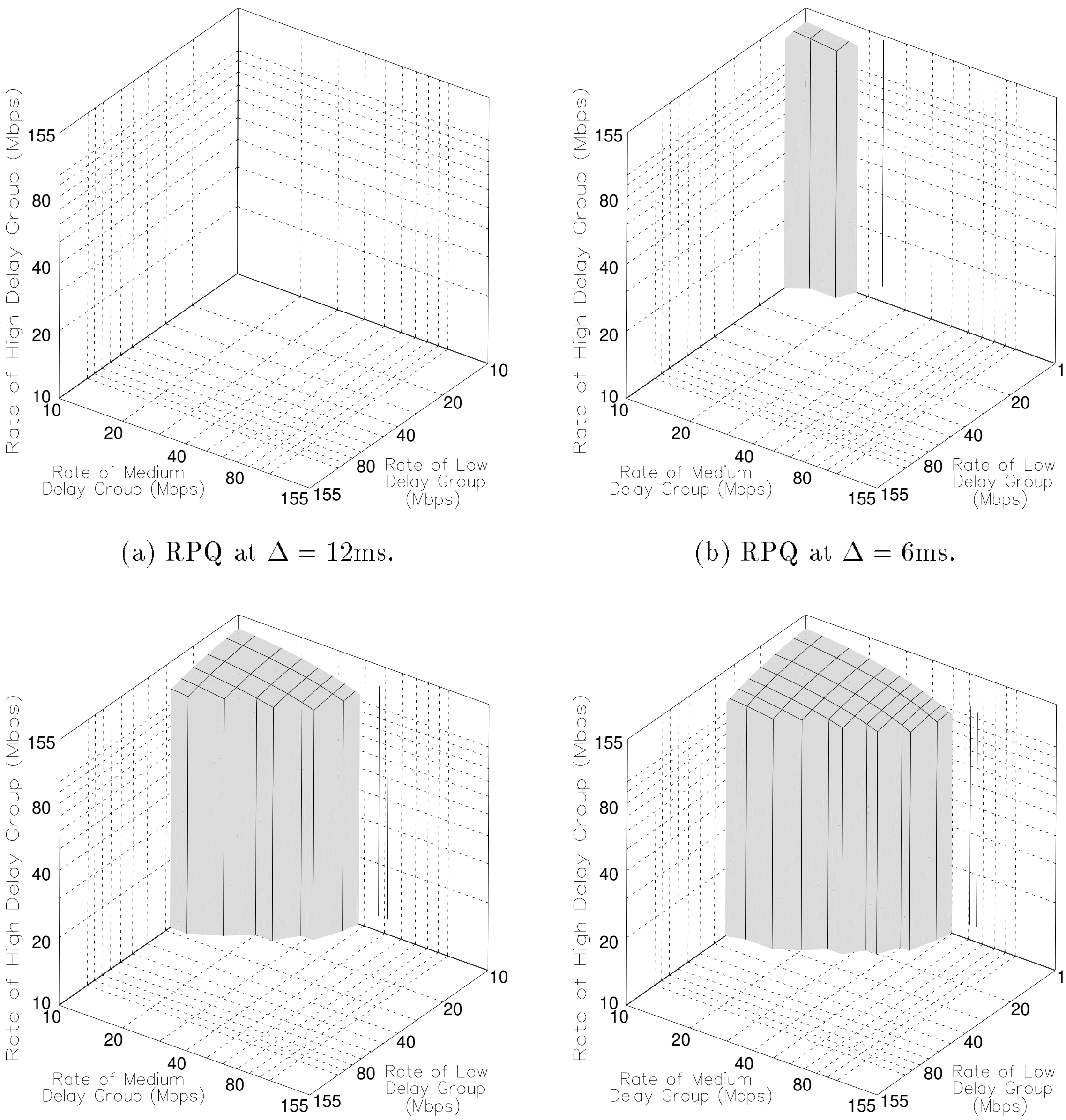
Table 5.1: Parameter set for scheduler with 155 Mbps transmission tate.

Figure 5.5 shows the schedulability regions for EDF and SP schedulers as well as a reference graph for evaluating the impact of a bounded-delay service. Figure 5.5(a) shows the schedulable region without delay constraints, i.e., $d_j = \infty$ for all j. In this case the schedulability condition is that the aggregate traffic rate cannot exceed the rate of the transmission link, that is, $\sum_{j=1}^{3} \rho_j < 155$ Mbps. This schedulable region will contain the schedulable region for all other packet schedulers.

In Figures 5.5(b) and 5.5(c), we depict the schedulable regions for EDF and SP packet schedulers, respectively. Since EDF is the optimal packet scheduler with respect to number of admissible connections, the region shown in Figure 5.5(b) will contain the region corresponding to any other packet scheduler. For the parameter sets considered, observe that the schedulable region for EDF is much larger than that for SP as shown in Figure 5.5(c).

In Figure 5.6, we illustrate schedulable regions of the RPQ scheduler for feasible rotation intervals in the range $\Delta = 1 - 12$ms. For this example, the number of queues that must be maintained for a particular choice of Δ is given by $1 + (36/\Delta)$, meaning that these examples use between 4 and 37 queues.

In Figures 5.6(a)-(f) note that the schedulable region increases as the rotation interval is decreased. The region for RPQ approaches that of EDF quickly and for $\Delta = 1$ms in Figure 5.6(f) the region is close to that of EDF. However, comparing the regions in Figures 5.6(a) and 5.6(b) with the the region for SP in Figure 5.5(c), we see that there are choices for Δ such that the efficiency of RPQ is inferior to that of SP. Since RPQ may



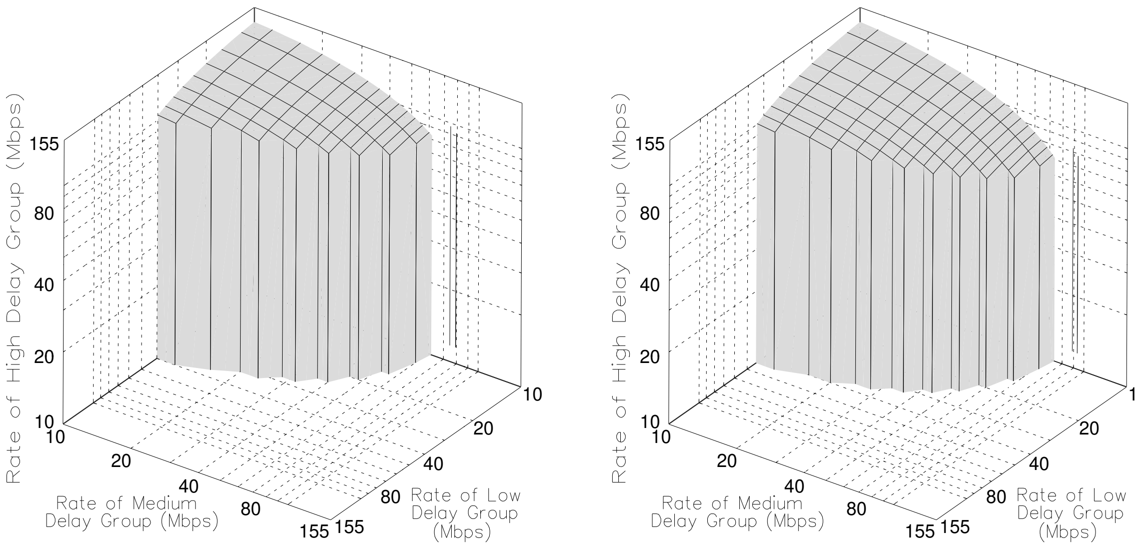
(a) RPQ at $\Delta = 12$ms.

(b) RPQ at $\Delta = 6$ms.

(c) RPQ at $\Delta = 4$ms.

(d) RPQ at $\Delta = 3$ms.

Figure 5.6: Schedulable regions for RPQ.

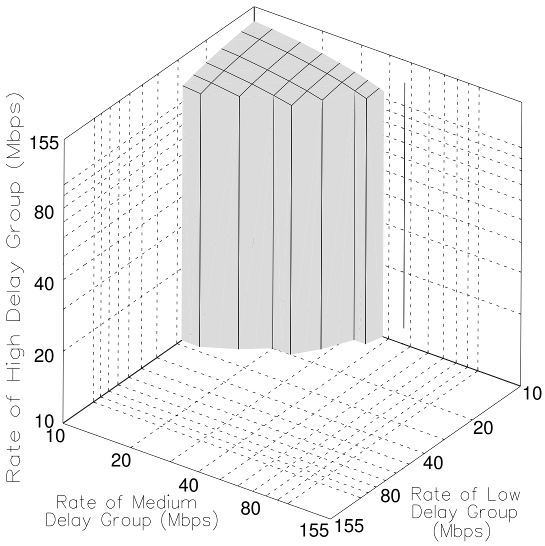


(e) RPQ at $\Delta = 2$ms. (f) RPQ at $\Delta = 1$ms.

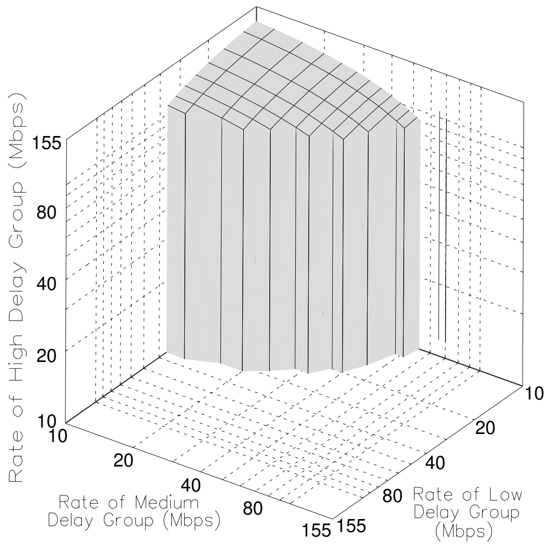
Figure 5.6: Schedulable regions for RPQ.

perform worse than SP, admission control mechanisms for RPQ should verify if it would be better to use SP rather of RPQ.

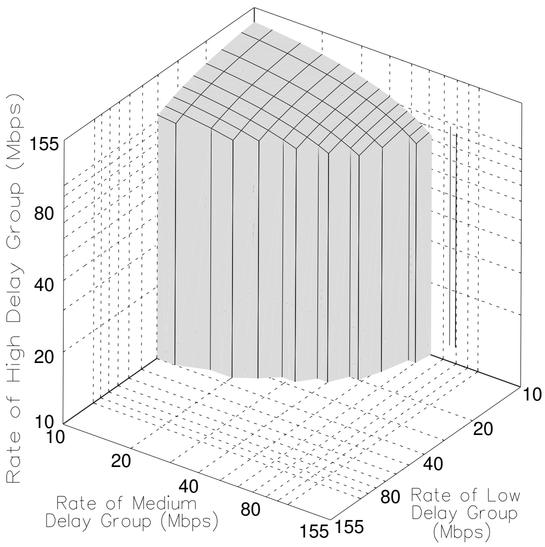
We next show in Figure 5.7 schedulable regions of the RPQ$^+$ packet scheduler for the same choices of rotation interval in the range $\Delta = 1 - 12$ms. The number of required queues for an RPQ$^+$ scheduler with rotation interval Δ in this example is $72/\Delta$. We first compare the RPQ$^+$ schedulable regions with the benchmark regions from Figure 5.5. Note that for all choices of rotation interval Δ, the RPQ$^+$ schedulable region is superior to that of SP in Figure 5.5(c). Even for $\Delta = 12$ms, the largest possible choice of RPQ$^+$ rotation interval for this example, the RPQ$^+$ schedulable region completely contains the SP region. Similar to RPQ, notice in Figures 5.7(a)-(f) that the schedulable region increases as the rotation interval is decreased, closely approximating EDF when $\Delta = 1$ms.



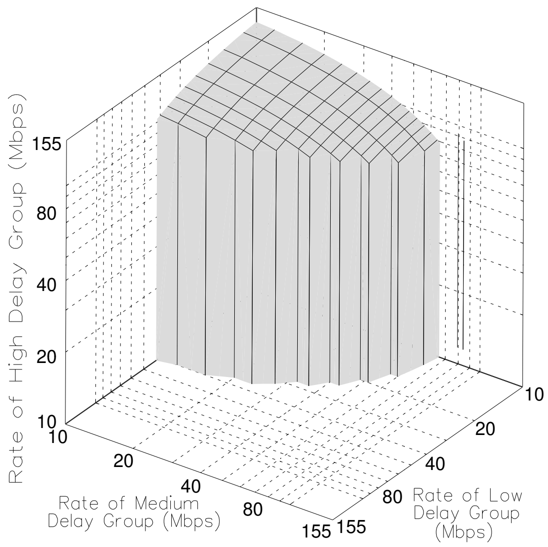
(a) RPQ$^+$ at $\Delta = 12$ms.



(b) RPQ$^+$ at $\Delta = 6$ms.

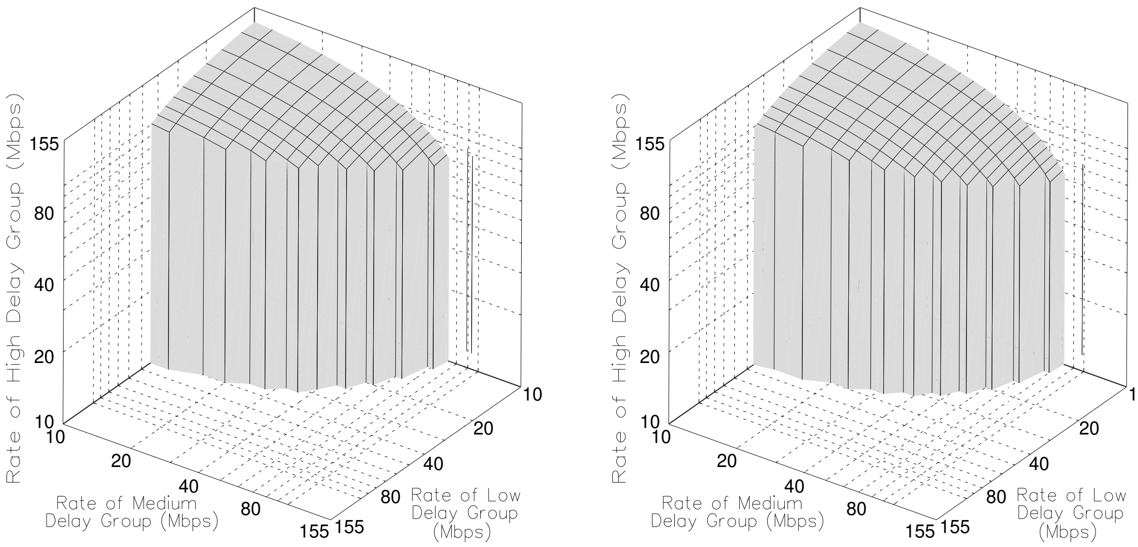


(c) RPQ$^+$ at $\Delta = 4$ms.



(d) RPQ$^+$ at $\Delta = 3$ms.

Figure 5.7: Schedulable regions for RPQ$^+$.



(e) RPQ$^+$ at $\Delta = 2$ms. (f) RPQ$^+$ at $\Delta = 1$ms.

Figure 5.7: Schedulable regions for RPQ$^+$.

Comparing the schedulability regions for RPQ$^+$ in Figure 5.7 with those for the RPQ scheduler in Figure 5.6, note that for all choices of Δ, the RPQ$^+$ scheduler achieves a larger schedulability region that the corresponding RPQ scheduler. Thus, for a given Δ, RPQ$^+$ achieves a higher efficiency.

Figure 5.8 summarizes the volumes from the schedulable regions in the previous figures by condensing the information into a single two-dimensional graph. In this graph we show results for the EDF, SP, and RPQ$^+$ schedulers considered in the previous figures, and we also include results for the RPQ scheduler. For a packet scheduler Σ, we let $V^{\Sigma}(\Delta)$ denote the volume of its schedulable region with rotation interval Δ.[3] Letting V^{∞} denote the volume of the schedulable region without deadlines shown in Figure 5.5(a), we use for evaluation

[3] Since the EDF and SP packet schedulers do not employ Δ, both $V^{SP}(\Delta)$ and $V^{EDF}(\Delta)$ are independent of Δ.

the ratio of $V^{\Sigma}(\Delta)$ and V^{∞} expressed as a percentage, that is:

$$\frac{V^{\Sigma}(\Delta)}{V^{\infty}} \cdot 100\%$$

We plot the resulting values for a packet scheduler as a function of Δ. For example, the value 42.1% for EDF in the figure can be interpreted as follows: the volume contained in the EDF schedulable region contains 42.1% of the volume of the region in Figure 5.5(a).

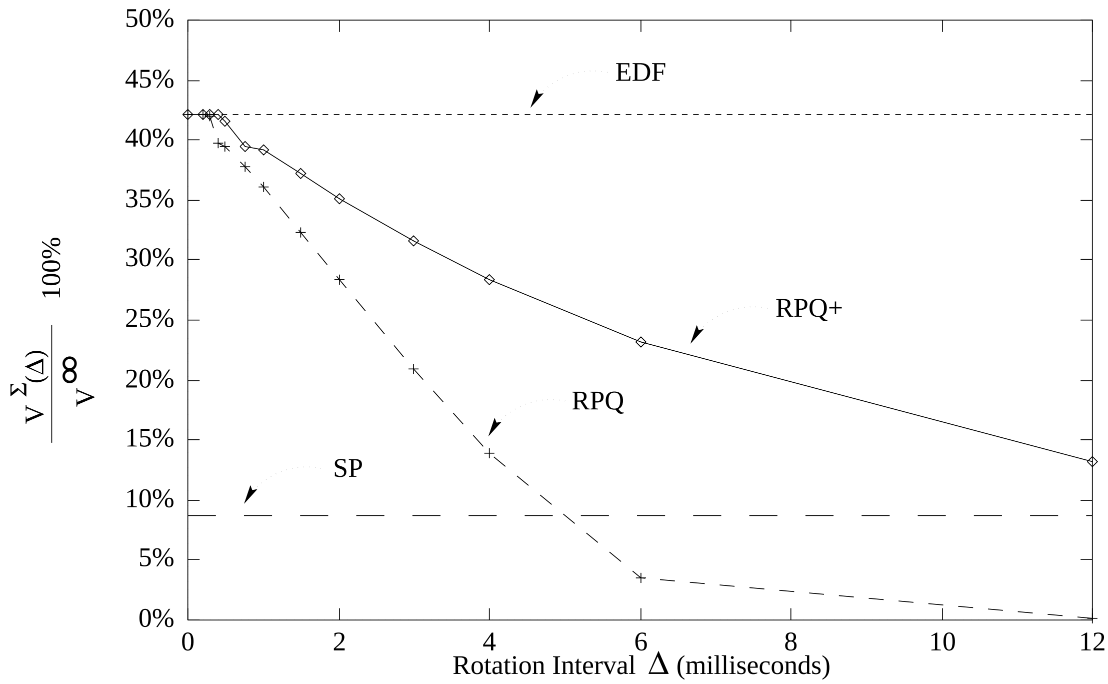


Figure 5.8: Summary of utilizations for packet schedulers. For each packet scheduler Σ, all values are reported as the ratio of $V^{\Sigma}(\Delta)$, i.e., the volume of the schedulability region of Σ with rotation interval Δ, and V^{∞}, i.e., the volume of the schedulability region as shown in Figure 5.5(a).

Figure 5.8 includes all rotation intervals Δ from Figures 5.6 and 5.7 as well as several additional values. Note in the figure that RPQ$^+$ achieves a schedulability region identical to EDF for $\Delta \leq 0.4$ms, while the RPQ schedulable region is identical to EDF for $\Delta \leq 0.2$ms.

Also observe that for this example the RPQ scheduler achieves an efficiency superior to SP only for $\Delta \leq 4$ms.

5.6.2 MPEG Example

In this experiment all traffic characterizations are obtained from publicly-available traces of MPEG video [89]. We use two MPEG traces for the evaluation: a thirty-minute segment of the James Bond entertainment movie *Goldfinger* (*"Bond"*) and 200 seconds of a video conference recorded using a set top camera ("*Settop*"). Both traces were encoded in software at 24 frames/second with frame size 384x288 and frame pattern IBBPBBPBBPBB.

We again consider a packet scheduler that operates at 155 Mbps, and we assume that all traffic is packetized in 53-byte ATM cells with a payload of 48 bytes each. We use the so-called *empirical envelope* of a video sequence to characterize its traffic, where the empirical envelope E^* of a sequence with traffic A is given by [17, 103]:

$$E^*(t) = \sup_{\tau \geq 0} A[\tau, \tau + t] \qquad \forall t \geq 0 \tag{5.29}$$

The empirical envelope is the tightest traffic characterization available for a video sequence and, when used with admission control, will result in the admission of a maximal number of connections. By using empirical envelopes for traffic characterization, we can determine the highest efficiency that can be achieved by a given packet scheduler.

Similar to the previous experiment, we consider two connection groups, one group consisting solely of *Bond* connections, and the second consisting solely of *Settop* connections. All connections in the same group have identical delay bounds: $d_{Settop} = 100$ms and $d_{Bond} = 200$ms.

Figure 5.9 illustrates the number of connections that can be supported at their delay constraints for the EDF, SP, RPQ, and RPQ$^+$ schedulers as well as for a peak-rate allocation scheme.[4] Figure 5.9(a) shows results for the RPQ scheduler, while Figure 5.9(b) shows

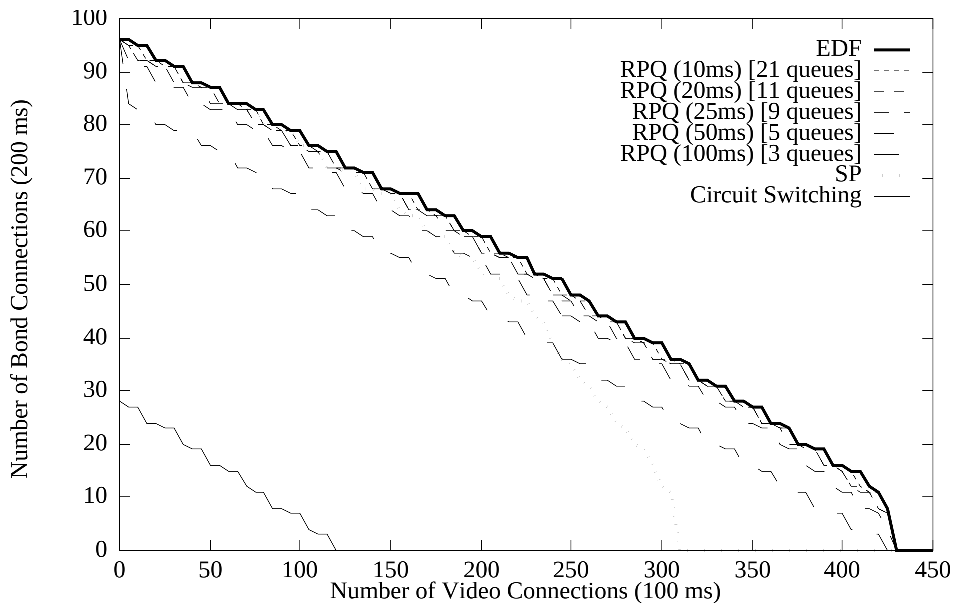
[4] The peak rate of a connection is defined as the ratio of the largest-sized frame and the constant interframe time.

results for RPQ$^+$. In both figures we use a bold solid line for EDF, a bold dotted line for SP, a thin solid line for the peak-rate allocation scheme, and a series of dashed lines for RPQ and RPQ$^+$ at different values of rotation interval Δ. For the RPQ and RPQ$^+$ curves we show the value of Δ and the number of required FIFOs. For each packet scheduler, we plot the maximum number of admissible *Bond* connections as a function of the number of *Settop* connections. For example, all packet schedulers (except the peak-rate scheme) can support 96 *Bond* connections if there are no *Settop* connections at the switch, and EDF can simultaneously support 60 *Bond* and 200 *Settop* connections.

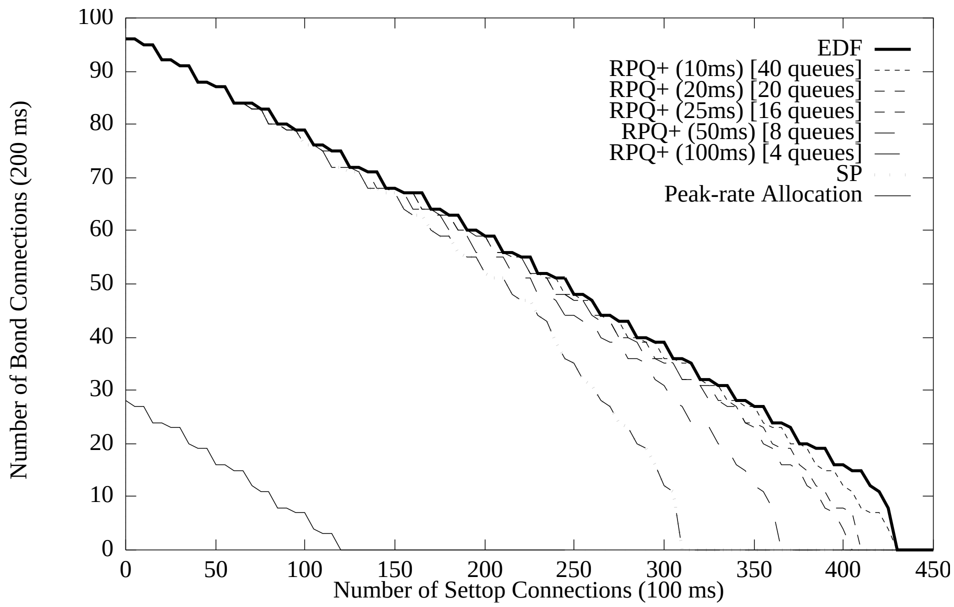
We observe in the figure that all of the packet schedulers can admit many more connections than the peak-rate allocation. Additionally, EDF is superior to SP when the number of higher-priority *Settop* connections is large. We observe in Figure 5.9(a) that RPQ is inferior to both EDF and SP when the number of high-priority connections is small. Note in Figure 5.9(b) that RPQ$^+$ is identical to SP for Δ = 100ms, and smaller values of Δ result in higher efficiency. For Δ = 10ms, a point of operation requiring 40 FIFOs, RPQ$^+$ closely approximates EDF.

5.7 Summary and Remarks

In this chapter we presented the novel Rotating-Priority-Queues$^+$ (RPQ$^+$) packet scheduler. On the one hand, we showed that RPQ$^+$ can achieve a network utilization that approximates that of Earliest-Deadline-First (EDF), the optimal scheduler that can support the tightest delay guarantees in a bounded-delay service. On the other hand, we showed that RPQ$^+$ can be implemented in shared-memory architectures with low overhead costs that make it practical for use in high-speed networks. The RPQ$^+$ scheduler places arriving packets into prioritized FIFO queues based on their delay constraints, and these queues are modified ("rotated") to increase the priority of waiting packets. RPQ$^+$ relies on a so-called rotation interval Δ that determines the frequency of these queue manipulations. When Δ is infinite, i.e., queues are never rotated, RPQ$^+$ behaves exactly the same as a SP scheduler and the



(a) RPQ



(b) RPQ^{+}

Figure 5.9: Schedulable regions for MPEG video traces.

two packet schedulers admit the same number of connections; as Δ is reduced, the number of admissible connections increases, approaching that of EDF in the limit. RPQ$^+$ can be implemented similar to SP except for the queue rotations which can be performed by manipulating pointers without moving any queued packets. We presented exact schedulability conditions for RPQ$^+$ that can be used for admission control mechanisms in a bounded delay service. We finally presented experiments that included MPEG traffic sources to illustrate that RPQ$^+$ can closely approximate EDF even for large values of Δ.

6

Conclusions and Future Work

6.1 Conclusions

Future integrated-services networks will support a variety of applications that differ widely in terms of both their traffic characteristics and quality-of-service requirements. The most demanding of these applications require a *bounded-delay service* that provides worst-case guarantees on network latencies.

In order to provide worst-case guarantees, the network must implement a resource reservation scheme to ensure the availability of resources such as bandwidth and buffer space for supporting the delay constraints of all traffic. A resource reservation scheme allows the network to quantify resource requirements of connections before they are established and to mathematically verify that all packets will be delivered within their delay constraints. Therefore, network resources are not shared as in traditional packet-switched networks and are rather allocated to individual connections. The number of admissible connections, and hence the utilization of network resources, is limited by two key components: the *traffic characterization* method which determines the resource requirements of each connection and the *packet scheduling discipline* that determines how resources are used at network switches. In this dissertation, we considered the design of novel traffic characterization methods and packet scheduling disciplines that can achieve high network utilization and are practical for use in high-speed networks.

6.1.1 Traffic Characterization

We presented a method for traffic characterization that determines accurate and practical characterizations for stored MPEG-compressed video. Our characterization method is based on approximating the empirical envelope of a video sequence which is the most accurate of all deterministic traffic characterizations. The empirical envelope has two drawbacks which prohibit its use as a traffic characterization: (1) it uses a large number of parameters which are computationally expensive to produce, and (2) it cannot be easily policed by traffic policing mechanisms. We presented a two-step method that addresses both of these problems. In the first step, we showed how to reduce the number of parameters needed for traffic characterization. We next used this reduced parameter set to determine a traffic characterization that can be policed by a small number of leaky bucket policing mechanisms. We used traces of MPEG-compressed video to demonstrate the effectiveness of our method. The outcomes of this research include the following:

- Only a small number (i.e., as few as 200 out of a total 40,000) of parameters of the empirical envelope are needed to produce an accurate traffic characterization for VBR video. Since these parameters can be generated quickly, it may be possible to determine accurate video characterizations in real time with our technique.

- Using our algorithm for parameter selection, we demonstrated that, contrary to conventional wisdom, using as few as four leaky buckets results in a characterization almost as precise as the empirical envelope.

- In our empirical evaluation, we evaluated our characterization method with other schemes from the literature and showed that our method was superior to all other known schemes for selecting leaky bucket parameters in terms of achievable network utilization.

We also devised a novel scheme that allows connections to dynamically renegotiate their traffic characterization. Although other renegotiation schemes exist in the literature, the scheme presented here is the only one that can be used in networks with a bounded-delay service. We demonstrated that our renegotiation scheme yields increases in network utilization of 20-35%.

6.1.2 Packet Scheduling

We presented the design, analysis, and evaluation of a novel scheduling discipline called Rotating-Priority-Queues$^+$ (RPQ$^+$). The RPQ$^+$ scheduler was motivated by the need for an effective scheduling discipline that can be implemented with low overhead costs. Although the Earliest-Deadline-First (EDF) is optimal in terms of achievable network utilization, implementing EDF requires complex sorting operations which are prohibitively expensive at high speeds. The RPQ$^+$ scheduling discipline can be viewed as an approximation of EDF that requires simple operations independent of the number of queued packets. The RPQ$^+$ scheduler is distinguished as follows:

- Different from other time-dependent schedulers, RPQ$^+$ does not require the sorting of individual packets, and all operations of the scheduler are independent of the number of queued packets. The only additional overhead of RPQ$^+$ as compared to a simple priority scheduler is a relabeling of FIFO queues.

- We presented an implementation of RPQ$^+$ for shared-memory architectures in which FIFO queue relabeling mentioned above can be implemented with low overhead costs using simple pointer arithmetic. With such an implementation RPQ$^+$ becomes attractive for use in high-speed networks.

- We derived exact admission control tests for RPQ^+, and so RPQ^+ can be used in networks that provide QoS guarantees. Using these admission control tests, we proved that RPQ^+ is always more efficient than SP and can approximate the optimal EDF scheduler with arbitrary precision. RPQ^+ is the only known packet scheduler with these properties that approximates EDF.
- We used MPEG video sources to show that RPQ^+ can closely approximate EDF using few FIFO queues.

6.2 Future Work

In this dissertation, we have presented a novel traffic characterization method for stored VBR video as well as a new packet scheduling discipline, and we demonstrated that both of these methods compare favorably with other schemes from the literature. Future directions of our traffic characterization work include applying our method to other types of VBR video, while work in scheduling includes a variant of RPQ^+ that would reduce the number of required queues as well as implementation. These directions are summarized below:

- In our experiments, we considered traffic characterizations of actual MPEG video sequences. However, recent research indicates that shaping VBR video traffic to reduce its burstiness may result in increased network utilization over multihop routes (e.g., [39, 92]). It should be straightforward to apply our characterization method to shaped video sequences, although some modifications may be required since shaped traffic is smoother than unshaped traffic.
- Our characterization method requires complete knowledge of a traffic source to determine its traffic characterization and is therefore appropriate only for stored traffic sources such as video-on-demand sequences. An important challenge is to obtain a worst-case characterization for "live" traffic sources. We note that some prior knowledge of the video sequence must be available to compute a characterization

for live video. However, one may be able to use information about the encoding scheme or recorded traces of similar video sequences to generate a reasonable traffic characterization that is more accurate than the peak traffic rate.

- Although the RPQ$^+$ packet scheduling provides an effective approximation of EDF, it may require the maintenance of a large number of logical queues. Consider for example a system that supports two delay bounds: 1 millisecond and 1 second. In this case, we must select $\Delta \leq 1$ms, and a minimum of 2000 queues are needed to support delay bounds as large as 1 second.

  The large number of required queues stems from the laxity range associated with each queue. (Recall that the laxity of a packet is the time remaining until a deadline violation.) In RPQ$^+$, the range of all laxities is partitioned into fixed-sized "bins," and each FIFO queue has a *fixed* laxity range of size Δ; for example, FIFO 1 holds packets with laxities in the range $[0, \Delta]$, FIFO 2 holds packets with laxities $[\Delta, 2\Delta]$, and a general FIFO p holds packets with laxities $[(p-1)\Delta, p\Delta]$. Thus, an RPQ$^+$ scheduler requires D/Δ FIFO queues to support a maximum delay bound of D. However, by allowing *variable* laxity ranges, one can design a scheduling discipline that trades off the accuracy of approximating EDF for a smaller number of FIFO queues. For example, consider a scheme where the laxity range increases exponentially: FIFO p holds packets with laxities in the range $[2^{p-1}\Delta, 2^p\Delta]$. In this case, a maximum delay bound of D could be supported with only $\log(D/\Delta)$ FIFO queues. Thus, this variant could use $O(\log P)$ FIFOs to support the same delay bound range as an RPQ$^+$ scheduler with $O(P)$ FIFOs.

- We have shown that the operations of an RPQ$^+$ scheduler are simple enough to be implemented at high speeds. However, practical issues and limitations should be further explored through simulation and implementation.

Bibliography

[1] ATM Forum, ATM Forum Traffic Management Specification Version 4.0, Contribution 95-0013R11. March 1996.

[2] ATM Forum, ATM User-Network Interface Specification Version 3.0, Prentice-Hall, 1993.

[3] CCITT Draft Recommendation I.371, Traffic Control and Resource Management in B-ISDN. December 1991.

[4] ForeThough Bandwidth Management Version 1.0, January 1996. Fore Systems White Paper.

[5] A. Adas and A. Mukherjee. On Resource Management and QoS Guarantees for Long Range Dependent Traffic. In *Proc. IEEE Infocom*, pages 779–787, April 1995.

[6] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne. Real-Time Communication in Packet-Switched Networks. *Proceedings of the IEEE*, 81(1):122–139, January 1994.

[7] F. Baker, R. Guerin, and D. Kandlur. Specification of Committed Rate Quality-of-Service. IETF Internet-Draft, June 1996.

[8] A. Banerjea, D. Ferrari, B. A. Mah, M. Moran, D. C. Verma, and H. Zhang. The Tenet Realttime Protocol Suite: Design, Implementation, and Experiences. *IEEE/ACM Transactions on Networking*, 4(1):1–10, February 1996.

[9] J. C. R. Bennett and H. Zhang. WF^2Q: Worst-case Fair Weighted Fair Queueing. In *Proc. IEEE Infocom '96*, pages 120–128, March 1996.

[10] A. W. Berger, A. E. Eckberg, T. C. Hou, and D. M. Lucantoni. Performance Characterizations of Traffic Monitoring, and Associated Control, Mechanisms for Broadband 'Packet' Networks. In *Proc. IEEE Globecom*, pages 350–353, December 1990.

[11] N. Berry. Asynchronous Transfer Mode (ATM) Switch Technology and Vendor Survey. Technical Report NAS-95-001, National Aeronautics and Space Administration, January 1995.

[12] P. E. Boyer, F. M. Guillemin, M. J. Servel, and J.-P. Coudreuse. Spacing Cells Protects and Enhances Utilization of ATM Network Links. *IEEE Network*, 6(5):38–49, September 1992.

[13] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF RFC 1633, July 1994.

[14] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. IETF Internet-Draft, May 1996.

[15] R. Brown. Calendar Queues: A Fast $O(1)$ Priority Queue Implementation for the Simulation Event Set Problem. *Communications of the ACM*, 21(10):1220–1227, October 1988.

[16] A. Campbell, G. Coulson, and D. Hutchison. A Quality of Service Architecture. *Computer Communications Review*, 24(2):6–27, April 1994.

[17] C.-S. Chang. Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[18] S. Chong and S. Li. (σ, ρ)-Characterization Based Connection Control for Guaranteed Services in High Speed Networks. In *Proc. IEEE Infocom '95*, pages 835–844, 1995.

[19] S. Chong, S. Li, and J. Ghosh. Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM. *IEEE Journal on Selected Areas in Communication*, 13(1):12–23, January 1995.

[20] D. D. Clark, S. Shenker, and L. Zhang. Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms. In *Proc. Sigcomm '92*, pages 14–26, August 1992.

[21] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1992.

[22] J. Crowcroft, Z. Wang, A. Smith, and J. Adams. A Rough Comparison of the IETF and ATM Service Models. *IEEE Network*, 9(6):12–16, November 1995.

[23] R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[24] R. L. Cruz. A Calculus for Network Delay, Part II: Network Analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[25] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. *Internetworking: Research and Experience*, 1:3–26, October 1990.

[26] L. Dittman, S. B. Jacobsen, and K. Moth. Flow Enforcement Algorithms for ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(3):343–350, April 1991.

[27] A. Elwalid, D. Mitra, and R. Wentworth. A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node. *IEEE Journal on Selected Areas in Communications*, 13(6):1115–1127, August 1995.

[28] D. Ferrari. Multimedia Network Protocols: Where are We? International Computer Science Institute, Berkeley, California, 1995.

[29] D. Ferrari. Client Requirements for Real-Time Communication Services. *IEEE Communications Magazine*, 28(11), November 1990.

[30] D. Ferrari. Real-Time Communication in an Internetwork. *Journal of High-Speed Networks*, 1(1):79–103, 1992.

[31] D. Ferrari and D. C. Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[32] N. R. Figueira and J. Pasquale. An Upper Bound on Delay for the VirtualClock Service Discipline. *IEEE/ACM Transactions on Networking*, 3(4):399–408, August 1995.

[33] N. R. Figueira and J. Pasquale. Leave-in-Time: A New Service Discipline for Real-Time Communications in a Packet-Switching Network. In *Proc. ACM Sigcomm*, pages 207–218, August 1995.

[34] V. Frost and B. Melamed. Traffic Modelling for Telecommunications Networks. *IEEE Communications Magazine*, 32(3):70–81, March 1994.

[35] D. Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):305–313, April 1991.

[36] M. W. Garrett. A Service Architecture for ATM: From Applications to Scheduling. *IEEE Network*, 10(3):6–14, May/June 1996.

[37] M. W. Garrett and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proc. ACM Sigcomm '94*, pages 269–280, August 1994.

[38] L. Georgiadis, R. Guerin, and A. Parekh. Optimal Multiplexing on a Single Link: Delay and Buffer Requirements. In *Proc. IEEE Infocom '94*, pages 524–532, June 1994.

[39] L. Georgiadis, R. Guerin, V. Peris, and K. N. Sivarajan. Efficient Network QoS Provisioning Based on per Node Traffic Shaping. In *Proc. IEEE Infocom '96*, pages 102–110, March 1996.

[40] S. J. Golestani. A Stop-and-Go Queueing Framework for Congestion Management. In *ACM Sigcomm '90*, pages 8–18, September 1990.

[41] S. J. Golestani. A Framing Strategy for Congestion Management. *IEEE Journal on Selected Areas In Communications*, 9(7):1064–1077, September 1991.

[42] S. J. Golestani. A Self-Clocked Fair Queueing Scheme for Broadband Applications. In *Proc. IEEE Infocom '94*, pages 636–646, June 1994.

[43] P. Goyal, S. Lam, and H. M. Vin. Determining End-to-End Delay Bounds in Heterogeneous Networks. In *Proc. 5th Intl. Workshop on Network Operating System Support for Digital Audio and Video*, pages 287–298, April 1995.

[44] M. Grossglauer, S. Keshav, and D. Tse. RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic. In *Proc. ACM Sigcomm '95*, pages 219–230, August 1995.

[45] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and its Applications to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas In Communications*, 9(7):968–981, September 1991.

[46] F. Guillemin, C. Rosenberg, and J. Mignault. On Characterizing an ATM Source via the Sustainable Cell Rate Traffic Descriptor. In *Proc. IEEE Infocom*, pages 1129–1136, April 1995.

[47] D. Heyman, A. Tabatabai, and T. V. Lakshman. Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks. *IEEE Transactions on Circuits Syst. Video Technol.*, 2(1), March 1992.

[48] D. P. Heyman and T. V. Lakshman. Source Models for VBR Broadcast-Video Traffic. In *Proc. IEEE Infocom '94*, pages 664–671, June 1994.

[49] J. M. Hyman, A. A. Lazar, and G. Pacifici. Real-Time Scheduling with Quality of Service Constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, September 1991.

[50] M. R. Ismail, I. E. Lambadaris, M. Devetsikiotis, and A. R. Kaye. Modelling Prioritized MPEG Video Using TES and a Frame Spreading Strategy for Transmission in ATM Networks. In *Proc. IEEE Infocom '95*, pages 762–770, 1995.

[51] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks. In *Proc. Sigcomm '96*, pages 2–13, August 1995.

[52] C. R. Kalmanek, H. Kanakia, and S. Keshav. Rate Controlled Servers for Very High-Speed Networks. In *Proc. Globecom '90*, pages 12–20, December 1990.

[53] G. Kesidis, J. Walrand, and C.-S. Chang. Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources. *IEEE/ACM Transactions on Networking*, 1(4):424–428, August 1993.

[54] E. W. Knightly. H-BIND: A New Approach to Providing Statistical Performance Guarantees to VBR Traffic. In *Proc. IEEE Infocom '96*, pages 1091–1099, March 1996.

[55] E. W. Knightly. *Traffic Models and Admission Control for Integrated Services Networks*. PhD thesis, University of California - Berkeley, May 1996.

[56] E. W. Knightly and P. Rossaro. Effects of Smoothing on End-to-End Performance Guarantees for VBR Video. In *Proc. International Symposium on Multimedia Communications and Video Coding*, October 1995.

[57] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic. In *Proc. ACM Sigmetrics '95*, pages 98–107, May 1995.

[58] E. W. Knightly and H. Zhang. Traffic Characterization and Switch Utilization Using Deterministic Bounding Interval Dependent Traffic Models. In *Proc. IEEE Infocom '95*, pages 1137–1145, April 1995.

[59] M. Krunz and H. Hughes. A Traffic Model for MPEG-Coded VBR Streams. In *Proc. ACM Sigmetrics '95*, pages 47–55, May 1995.

[60] J. F. Kurose. On Computing Per-Session Performance Bounds in High-Speed Multi-Hop Computer Networks. In *Proc. 1992 ACM Sigmetrics and Performance '92*, pages 128–139, June 1992.

[61] J. F. Kurose. Open Issues and Challenges in Providing Quality of Serviice Issues in High Speed Networks. *Computer Communications Review*, 23(1):6–15, January 1993.

[62] S. S. Lam, S. Chow, and D. K. Y. Yau. An Algorithm for Lossless Smoothing of MPEG Video. In *Proc. ACM Sigcomm '94*, pages 281–293, August 1994.

[63] S. S. Lam and G. G. Xie. Burst Scheduling: Architecture and Algorithm for Switching Packet Video. In *Proc. IEEE Infocom*, pages 940–950, April 1995.

[64] S. S. Lam and G. G. Xie. Group Priority Scheduling. In *Proc. IEEE Infocom '96*, pages 1346–1356, March 1996.

[65] A. Lazar, G. Pacifici, and D. Pendarakis. Modeling Video Sources for Real-Time Scheduling. *Multimedia Systems*, 1:253–266, April 1994.

[66] A. A. Lazar, A. T. Temple, and R. Gidron. MAGNET II: A Metropolitan Area Network Based on Asynchronous Time Sharing. *IEEE Journal on Selected Areas in Communications*, 8(10):1582–1594, October 1990.

[67] J. P. Lehoczky and S. Ramos-Thuel. An Optimal Algorithm for Scheduling Soft-Aperiodic Tasks in Fixed-Priority Preemptive Systems. In *Real-Time Systems Symposium*, pages 110–123, December 1992.

[68] J. Y. T. Leung and J. Whitehead. On the Complexity of Fixed-Priority Scheduling of Periodic, Real-Time Tasks. *Performance Evaluation*, 2(4):237–250, December 1982.

[69] S. Q. Li, S. Chong, C. Hwang, and X. Zhao. Link Capacity Allocation and Network Control by Filtered Input Rate in High Speed Networks. In *Proc. IEEE Globecom '93*, pages 744–750, December 1993.

[70] J. Liebeherr. Multimedia Networks: Issues and Challenges. *IEEE Computer*, 28(4):68–69, April 1995.

[71] J. Liebeherr and D. Liao. A Service with Bounded Degradation in Quality-of-Service Networks. In *Proc. IEEE Infocom '95*, pages 1103–1110, April 1995.

[72] J. Liebeherr and D. E. Wrege. A Versatile Packet Multiplexer for Quality-of-Service Networks. In *Proc. 4th International Symposium on High-Performance Distributed Computing (HPDC-4)*, pages 148–155, August 1995.

[73] J. Liebeherr and D. E. Wrege. An Efficient Solution to Traffic Characterization of VBR Video in Quality-of-Service Networks. Technical Report TR-96-10, Department of Computer Science, University of Virginia, May 1996. Submitted.

[74] J. Liebeherr, D. E. Wrege, and D. Ferrari. Exact Admission Control in Networks with Bounded Delay Services. To appear: *IEEE/ACM Transactions on Networking*, December 1996.

[75] Y. Lim and J. E. Kobza. Analysis of a Delay-Dependent Priority Discipline in an Integrated Multiclass Traffic Fast Packet Switch. *IEEE Transactions on Communications*, 38(5):659–665, May 1990.

[76] C. L. Liu and J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment. *Journal of the ACM*, 20(1):46–61, January 1973.

[77] S. Low and P. Varaiya. A Simple Theory of Traffic and Resource Allocation in ATM. In *Proc. IEEE Globecom '91*, pages 1633–1637, 1991.

[78] J. M. McManus and K. W. Ross. Video on Demand over ATM: Constant-Rate Transmission and Transport. To appear: *IEEE Journal on Selected Areas in Communications*.

[79] B. Melamed and B. Sengupta. TES Modeling of Video Traffic. *IEICE Transactions on Communications*, E75-B(12):1292–1300, 1992.

[80] P. Pancha and M. El Zarki. Leaky Bucket Access Control for VBR MPEG Video. In *Proc. IEEE Infocom*, pages 796–803, April 1995.

[81] A. K. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, February 1992.

[82] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[83] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[84] J. M. Peha. *Scheduling and Dropping Algorithms to Support Integrated Services in Packet-Switched Networks.* PhD thesis, Stanford University, June 1991.

[85] J. M. Peha and F. A. Tobagi. Implementation Strategies for Scheduling Algorithms in Integrated-Services Packet-Switched Networks. In *Proc. IEEE Globecom '91*, pages 1733–1740, 1991.

[86] E. P. Rathgeb. Modeling and Performance Comparison of Policing Mechanisms for ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(4):325–334, April 1991.

[87] E. P. Rathgeb. Policing of Realistic VBR Video Traffic in an ATM Network. *International Journal of Digital and Analog Communications Systems*, 6:213–226, 1993.

[88] A. R. Reibman and A. W. Berger. Traffic Descriptors for VBR Video Teleconferencing Over ATM Networks. *IEEE/ACM Transactions on Networking*, 3(3):329–339, June 1995.

[89] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. Technical Report 101, Institute of Computer Science, University of Wurzburg, February 1995.

[90] C. Rosenberg and B. Lague. A Heuristic Framework for Source Policing in ATM Networks. *IEEE/ACM Transactions on Networking*, 2(4):387–397, August 1994.

[91] D. Saha, S. Mukherjee, and S. H. Tripathi. Carry-Over Round Robin: A Simple Cell Scheduling Mechanism for ATM Networks. In *Proc. IEEE Infocom '96*, pages 630–637, March 1996.

[92] J. Salehi, Z. Zhang, J. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In *Proc. ACM Sigmetrics '96*, pages 222–231, May 1996.

[93] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 1889, January 1996.

[94] S. Shenker and L. Breslau. Two Issues in Resource Establishment. In *Proc. Sigcomm '95*, pages 14–26, Cambridge, MA, August 1995.

[95] S. Shenker and J. Wroclawski. Network Element Service Specification Template. IETF Internet-Draft, June 1995.

[96] M. Shreedhar and G. Varghese. Efficient Fair Queueing using Deficit Round Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, June 1996.

[97] J. A. Stankovic and K. Ramamritham (editors). *Hard Real-Time Systems*. IEEE Computer Society Press, 1988.

[98] D. Stiliadis and A. Varma. Design and Analysis of Frame-based Fair Queueing: A New Traffic Scheduling Algorithm for Packet-Switched Networks. In *Proc. ACM Sigmetrics '96*, pages 104–115, May 1996.

[99] C. Topolcic. Experimental Internet Stream Transport Protocol: Version 2 (ST-II). IETF RFC 1190, October 1990.

[100] J. S. Turner. New Directions in Communications (or Which Way to the Information Age?). *IEEE Communications Magazine*, 25(8):8–15, October 1986.

[101] J. S. Turner. Managing Bandwidth in ATM Networks with Bursty Traffic. *IEEE Network*, 6(5):50–58, September 1992.

[102] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing Delay Jitter Bounds in Packet Switching Networks. In *Proc. Tricomm '91*, Chapel Hill, North Carolina, April 1991.

[103] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr. Deterministic Delay Bounds for VBR Video in Packet-Switching Networks: Fundamental Limits and Practical Tradeoffs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

[104] D. E. Wrege and J. Liebeherr. A Near-Optimal Packet Scheduler for QoS Networks. Technical Report TR-96-12, Department of Computer Science, University of Virginia, June 1996. Submitted.

[105] D. E. Wrege and J. Liebeherr. Video Traffic Characterization for Multimedia Networks with a Deterministic Service. In *Proc. IEEE Infocom '96*, pages 537–544, March 1996.

[106] J. Wroclawski. Specification of the Controlled-Load Network Element Service. IETF Internet-Draft, June 1996.

[107] G. G. Xie and S. S. Lam. Delay Guarantee of Virtual Clock Server. *IEEE/ACM Transactions on Networking*, 3(6):683–689, December 1995.

[108] H. Zhang. *Service Disciplines for Packet-Switching Integrated-Services Networks*. PhD thesis, University of California - Berkeley, November 1993.

[109] H. Zhang. Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines. *Computer Communications: Special Issue on System Support for Multimedia Computing*, 18(10), October 1995.

[110] H. Zhang. Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. *Proceedings of the IEEE*, 83(10):1374–1399, October 1995.

[111] H. Zhang and D. Ferrari. Rate-Controlled Static-Priority Queueing. In *Proc. IEEE Infocom '93*, pages 227–236, April 1993.

[112] H. Zhang and S. Keshav. Comparison of Rate-Based Service Disciplines. In *Proc. Sigcomm '91*, pages 113–121, Zurich, Switzerland, September 1991.

[113] H. Zhang and E. W. Knightly. Providing End-to-End Statistical Performance Guarantees with Bounding Interval Dependent Stochastic Models. In *Proceedings of ACM Sigmetrics*, pages 211–220, May 1994.

[114] H. Zhang and E. W. Knightly. A New Approach to Support Delay-Sensitive VBR Video in Packet-Switched Networks. In *Proc. 5th Intl. Workshop on Network Operating System Support for Digital Audio and Video*, pages 275–286, April 1995.

[115] L. Zhang. *A New Architecture for Packet Switched Networks.* PhD thesis, Massachusetts Institute of Technology, July 1989.

[116] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proc. ACM Sigcomm*, pages 19–29, September 1990.

[117] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, 7(5):8–18, September 1993.

[118] Q. Z. Zheng and K. G. Shin. On the Ability of Establishing Real-Time Channels in Point-to-Point Packet Switching Networks. *IEEE Transactions on Communications*, 42(3):1096–1105, March 1994.