Providing Quality of Service in Packet Switched Networks[1]

Don TOWSLEY
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
U.S.A.

Abstract

Increases in bandwidths and processing capabilities of future packet switched networks will give rise to a dramatic increase in the types of applications using them. Many of these applications will require guaranteed quality of service (QOS) such as a bound on the maximum end-to-end packet delay and/or on the probability of packet loss. This poses exciting challenges to network designers. In this paper we discuss the QOS requirements of different applications and survey recent developments in the areas of call admission, link scheduling, and the interaction between the provision of QOS and call routing and traffic monitoring and policing. We identify what some of the important issues are in these areas and point out important directions for future research efforts.

Keywords : quality of service, call admission, real-time services, link scheduling.

1 Introduction

Networking is evolving at a rapid pace these days. Only ten years ago, wide area network bandwidths were in the range of 56Kbs and less. Typical applications were electronic mail, file transfers and remote login. Present day wide area networks have bandwidths in the order of 45Mbs (NSFnet backbone) and applications include, in addition to those listed above, digitized voice and low rate video. In addition, there exist several experimental networks that are operating in the gigabit range. Work is underway on a wide variety of high bandwidth applications such as HDTV, medical diagnosis, multimedia conferencing, etc..

One of the burning issues in this evolution deals with the provision of adequate service for these new applications. Recent experience on the internet indicates that it is ill suited to handle time and loss sensitive applications such as voice and video. This is due not only to the inadequate bandwidth provided by the internet but, more importantly, because the internet does not provide the

[1] This work was supported in part by the National Science Foundation under grant NCR-9116183.

right support in the form of end-to-end protocols and switch scheduling policies. It is the aim of this paper to describe issues and problems that arise in providing quality of service (QOS) to applications along with several approaches that have been proposed and studied in the literature.

We will discover that the problem of providing quality of service to B-ISDN applications is complex and that it cannot be solved satisfactorily without also dealing with numerous other problems such as routing, congestion control, link scheduling, traffic shaping and monitoring, etc... We will briefly discuss the realtionship between these problems and that of providing quality of service. We will observe that the solutions to some of these problems such as traffic shaping and link scheduling are intricately related to the design of algorithms for providing QOS. Other problems, such as routing can be treated in a more detached manner.

The remainder of this paper is structured in the following way. Section 2 introduces a sample of B-ISDN applications focussing primarily on their QOS requirements. A discussion of how they are typically modelled and their salient workload parameters is also included in section 2. Section 3 describes the problem of providing QOS in greater detail including a discussion of a number of issues that must addressed by any network architecture supporting QOS guarantees. Section 4 will discuss the problems of traffic shaping and monitoring and will describe the rate control paradigm commonly included as a component of a network architecture that provides for QOS. A description and discussion of several link scheduling algorithms, which can be used to support applications having diffeent QOS requirements is found in section 5. The primary focus of the paper is on call admission which is the subject of section 6. Section 7 summarizes the main ideas and lists a number of directions for further research.

2 Applications and their QOS Requirements

Applications can be divided broadly into two classes, those without real-time constraints and those with. Traditional networking applications such as file transfer, electronic mail, and remote login do not have real-time constraints. The performance metrics of interest for these applications are typically average packet delay and throughput. They also require full reliability which is provided by high level end-to-end protocols.

Of more interest to us are those applications having real-time constraints. These include voice and video. Such applications are characterized by a bound, D, on the time, T, allowed to transmit a packet across the network. Thus, a deadline is associated with each packet and, if the packet reaches its destination after its deadline, it may be considered useless and discarded.

The following two QOS requirements have been proposed for real-time applications in the literature,

$$\begin{array}{lrcl} \text{(Q1)} & T & \leq & D, \\ \text{(Q2)} & \Pr[T > D] & < & \epsilon. \end{array}$$

The first of these metrics requires that the application suffer no packet loss. Clearly this is impossible to achieve given that network components can fail and communication links can corrupt packets. Hence, it is normally interpreted to mean that the application requires no losses beyond what may be introduced by component failures and link noise. Henceforth, we ignore link noise and component failures and assume a fully reliable network. Fortunately, the most common real-time applications, voice and video can tolerate some fraction of either lost or delayed packets, (approximately $10^{-6} - 10^{-2}$). Here losses occur as a result of buffer overflow.

For some applications, such as voice and video, there is no benefit to having packets arrive far ahead of their deadlines. This is because the receiver is required to store these packets until the deadline at which point the data can be played out. Thus the following QOS requirement has also been considered

$$\text{(Q3)} \qquad T_{max} - T_{min} \leq J.$$

Figure 1 illustrates the first and third criteria (Q1 and Q3).

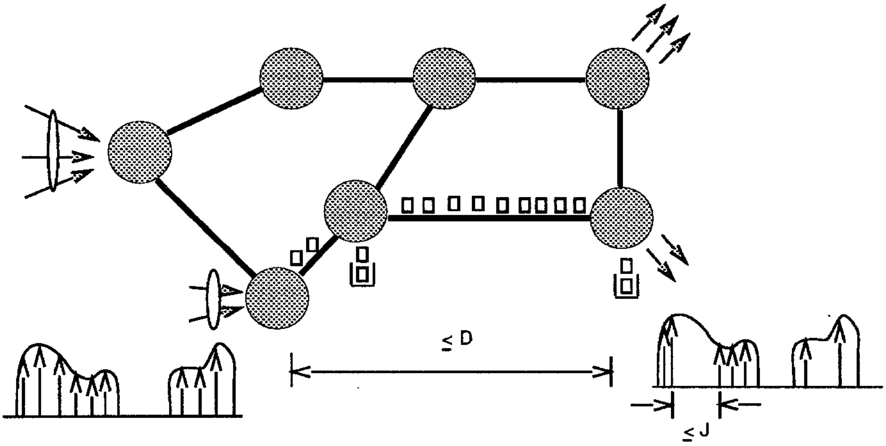

Figure 1: End-to-end delay and jitter bounds.

The fourth QOS requirement commonly treated in the literature is

$$\text{(Q4)} \qquad \Pr[\text{end-to-end packet loss}] < \epsilon$$

This may be appropriate for real-time applications such as voice operating on networks where packets that, because of their architecture, guarantee that any packet reaching its destinationhas an end-to-end delay less than D. It may also be appropriate for non-real-time applications as it bounds the number of retransmissions that are required to ensure lossless data transfer.

Based on these QOS requirements, we divide applications into three classes,

- deterministic (Q1, Q3),
- statistical (Q2, Q4), and
- best effort (no requirement).

Although, these are the QOS metrics typically considered in the literature on provision of QOS, they may not be the most appropriate metrics, [39, 40, 6, 43, 30, 5, 37]. For example, consider the requirement that packet loss not exceed 1% for a voice application. If the application ultimately transmits 100,000 packets, then there is a considerable difference in the user's perception of the quality of the voice if the first packet out of each group of 100 are lost rather than the first 100 packets out of each group of 10,000. Thus a number of papers propose QOS requirements based on intervals of time, e.g., 1% loss over a talkspurt in audio applications [40, 6], over a frame in video [43, 30, 5, 37]. Recent work [36] indicates that replacing interval QOS measures with stationary QOS measures can produce very poor results if, in fact interval measures are of interest. There has been little work performed to include these types of metrics as part of call admission algorithms. Tlre design of call admission algorithms to provide QOS guarantees is based on stationary metrics. Interval QOS metrics could then be satisfied by choosing overly stringent stationary connetion oriented requirements. Hence, we will focus on criteria Q1 – Q4 in the remainder of the paper.

We conclude this section with a discussion of workload characteristics of the applications envisioned for B-ISDN. Although a number of applications are characterized by a continuous bit rate (CBR, e.g., voice without silence detection), most applications use compression techniques in order to reduce their average bandwidth requirements. These applications produce bursty, highly correlated packet streams. Current traffic source models include Markov modulated arrival processes (e.g., [24]) and Markov modulated fluid processes, (e.g., [3]). In numerous cases, voice, medical images, and file transfers, they can be modeled by two state on-off models. If the process is in an off state, no packet is generated and if it is in an on state, packets are generated with at constant intervals of time. We will not discuss these models farther except to state that the typical statistics associated with these processes that are used in provision of QOS include x_{min}, the minimum interarrival time between packets, L_max, the maximum packet size, R_{peak}, the peak rate, b, the expected burst length, and u, the fraction of time the source is on. The problem of modeling packet streams for different applications is far from having been solved. Future work in this area will undoubtedly impact ways for providing QOS.

3 Overview of Providing QOS

QOS requirements are typically specified on an *end-to-end* basis. This imposes requirements on the hosts at each end as well as the network connecting them. We will only focus on the network QOS requirements in this paper (although many of the ideas could be applied to the hosts). QOS requirements are best satsisfied through connection oriented services. A connection consists of three phases, call setup, data transport, and call breakdown. In the preceding section, we discussed different potential data transport QOS requirements. However, the setup and breakdown phases can have their own requirements. For example, setup requirements typically include constraints on connection blocking probability and connection setup delay and breakdown requirements typically include constraints on breakdon delay. We will focus on the provision of QOS during the transport phase.

Call set up is concerned with answering the following question,

The question: Can the call be admitted so that its QOS requirements are met without violating the QOS requirements of any existing call in the network?

This involves choosing a physical path between the two end nodes. Next a call setup control packet is sent along this path to determine whether it can meet the QOS requirements without violating those of any existing call in the network. This involves checking at each node whether sufficient resources exist to do so. If the determination is made that each node can do so, then the resources are reserved and the call accepted.

We are concerned with answering "the question" with repect to a single physical path. Although path selection (routing) is extremely important in obtaining good performance, its solution is, to some extent, orthogonal to the design of good algorithms for provision of QOS on a single path. Various aspects of the routing problem and its relationship to the provision of QOS can be found in [26]. Solutions to the routing problem will undoubtedly borrow from routing in circuit-switched networks, [18].

Consider the problem of answering the question whether or not the QOS requirements of a new call can be satisfied while continuing to provide the QOS of other calls. In its general form, this is an extremely difficult question to answer as the admission of a call has the potential of affecting every session currently using the network. In order to mitigate this problem most approaches follow the *principle of nodal isolation*; namely that the behavior of the nodes on the path should be isolated from each other and from that of the remaining nodes in the network. This can be accomplished in one of two ways, 1) statically partitioning resources among connections, or 2) imposing local QOS requirements on each node so that each node on the path determines independently whether it can satisfy these local QOS requirements. We shall refer to the former as the *principle of connection isolation*and will examine approaches based on both as well as a hybrid. The reader is referred to [49] for further design principles for call admission algorithms.

In order to answer "the question", a connection must provide workload descriptors along with its QOS requirements. The previous section described some proposed descriptors and requirements. A successful call setup corresponds to a contract between the application and network. In order to prevent an application from breaking its part of the contract, i.e., that it will behave according to the description that it provided the network, the network should provide a traffic monitoring and policing mechanism at the edge of the network. An example of such a mechanism is the leaky bucket [46]. Further details are found in section 4. A second reason for monitoring the traffic of a connection would be to provide adaptivity. If the traffic characteristics of a connection change, then the network could note these changes and make use of them when setting up additional calls. This approach is taken in [22, 11].

Most of the work in this area is based on the underlying assumption that bursts generated by sources are typically small compared to the buffer capacity of each node in the network. Most of the approaches that we will focus on in this paper will not work well if bursts are comparable in size or larger than buffer sizes. If the buffer size is much smaller than the burst size, then the only solution may be to perform fast circuit switching at a burst level. An example of this approach is found in [47].

4 Traffic Shaping and Monitoring

As we have observed in section 3, a traffic shaping mechanism may be necessary to ensure that a source behaves according to the characterization that it provides the network at the time that it establishes its session. Traffic shaping mechanisms come in many different flavors. However, they are mostly variations of the *leaky bucket* rate control mechanism originally proposed by Turner, [46]. We briefly describe one variation which is used by a number of different proposed bandwidth allocation policies.

Briefly, a leaky bucket consists of a data buffer and a finite capacity token buffer. Packets enter the data buffer from the source. Tokens are generated deterministically at rate rho and immediately enter the token buffer. Whenever a packet containing p bits enters the data buffer and finds at least p tokens in the token buffer, the packet immediately is released to the network. Otherwise it queues up in the data buffer and waits until it is the oldest packet and p tokens have accumulated. At that time, the packet is allowed to enter the network. The capacity of the token buffer is σ and is used to control the burst length of the source. Last, a peak bandwidth enforcer is used to ensure that the peak bandwidth never exceeds C. The leaky bucket is illustrated in Figure 2.

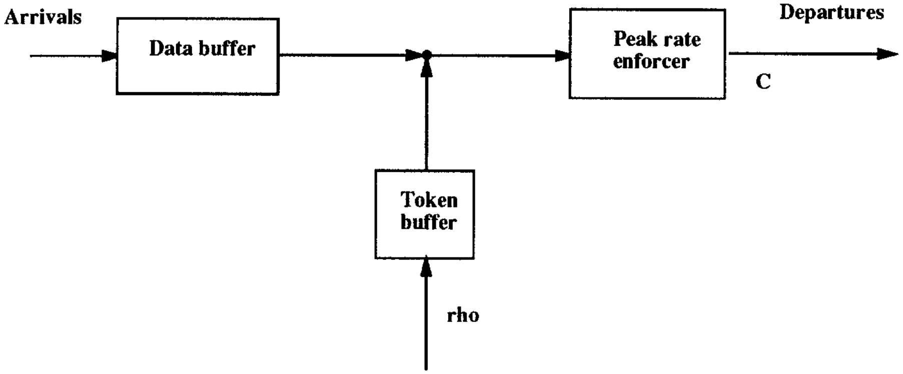

Figure 2: Leaky Bucket Functional Diagram.

As mentioned before, many variations of this mechanism exist. One variation, suited to ATM, allows packets to always enter the network without delay. However, if the packet arrives to find an insufficient number of tokens, it is marked before being released. Coupling this with a link scheduling policy that is allowed to drop marked packets in the case of congestion, can yield performance improvements over the standard leaky bucket, [16]

A source is said to be a (σ, ρ, C) *linearly bounded arrival process (LBAP)* if, when fed through a leaky bucket with parameters σ, ρ, and C, none of the packets ever incur a delay. The leaky bucket always guarantees that the network will see a (σ, ρ, C) LBAP. Often times C is taken to be ∞. In that case the peak rate parameter will be omitted, e.g., (σ, ρ) LBAP instead of (σ, ρ, ∞) LBAP.

Before ending this section, we mention that the leaky bucket has been the subject of many studies. Most have focussed on evaluating its performance, either in isolation, e.g., [44], or feeding one or more downstream queues, e.g., [41]. More recent work has focussed on formally stating and proving a number of burst reduction properties exhibited by this mechanism. These include burstiness exhibited by the departure process, or by its effects on delays and/or losses at downstream queues [35, 34].

5 Link Scheduling Policies

In a high speed network setting, link scheduling policies must allow different classes of applications having different quality of service requirements to share a link. In this section, we describe some of the issues that arise in designing multiclass scheduling policies and how they have been treated by network designers. The reader is referred to [23, 4] for additional details regarding the link scheduling policies described here as well as others.

We assume that the scheduling policy must deal with at least three classes of applications which differ according to the type of QOS that they require, *i)* deterministic, *ii)* statistical, and *iii)* best effort. Each of these may consist of additional subclasses. For the sake of discussion we assume there are only these three classes and that a policy schedules the link between S_k different sessions labelled $s = 1, \ldots, S_k$ where $k \in \{d, s, b\}$ denotes the class of application. Further, when describing how a policy schedules packets from a specific session, it is helpful to refer to the other servers on the path allocated to that session. Hence, we shall refer to the path for session s as $\Gamma_s = (j_{i,1}, ..., j_{i,n_s})$ where $J_{i,k}$ is the k-th server on the path. We shall refer to the server under consideration as l_s in this context.

A scheduling policy has to deal with several issues. First, to what extent will it isolate the effects of different classes of applications from each other? Second, to what extent will it isolate the effects of different sessions within the same class from each other? This suggests that a policy has a hierarchic structure. At the top level of the hierarchy is a mechanism for scheduling between different classes of sessions. At the second level will be a mechanism for scheduling sessions from the same class. This, of course may differ from class to class. If one of the classes is further subdivided into additional subclasses, then the policy may contain three or more levels.

The issue of whether or not to isolate sessions/classes from each other is an extremely important one. Traditional circuit switching is characterized by absolute session isolation; each session is given a fixed fraction of the bandwidth and is never aware of other sessions. Such isolation can be emulated in B-ISDN by a careful implementation of space division/ time division multiplexing where each session obtains a fraction of the bandwidth corresponding to its *peak rate*. This policy has the advantage that it simplifies the problem of making deterministic guarantees. However, it may provide very inefficient use of the bandwidth if the applications are bursty. Such a philosophy, with some modifications to provide flexibility lies behind three recently proposed policies, Hierarchical round robin (HRR) ([29]), Stop-and-Go (SG) ([20], and Weighted fair queueing (WFQ) ([13, 32]). We will describe these later in this section.

It is worth pointing out that the ATM standard provides for virtual paths [7, 8] between source-destination pairs. Briefly, a virtual path can be viewed as a dedicated bandwidth channel between a pair of nodes. Hence, it is a mechanism that can be used to isolate different classes of service from each other.

A second approach to session/class isolation is to assign static priorities to different sessions. Thus, for example priority could be given in decreasing order to deterministic, statistical, and best effort service. This appears at first sight to be a good solution as it appeals to our intuition regarding the relative importance and urgency of the three classes of services. However, as a general multiclass scheduling policy it has been found wanting in several respects. First, it is very inflexible. It has been observed in that policies that attempt to cooperatively share the bandwidth between statistical and best effort classes of service can provide better performance for the latter class of service with either no or marginal degradation in the QOS of the former class. Second, it may provide overkill to the class of service receiving the highest priority of service. For example, there is no benefit in transmitting a packet far ahead of its deadline. Instead, it may be possible to provide better performance to the other classes of service packets by delaying the packet's transmission for awhile and allocating the link instead to these other classes of service. These have been illustrated in [10].

It is important to point out that most proposed algorithms provide higher priority to packets generated by deterministic and statistical applications than to best effort packets.

There have been some attempts to develop high level scheduling policies that attempt to share the link cooperatively between different classes of service rather than to isolate them, [27]. We will survey these policies later in this section.

Last, a scheduler has to be chosen to schedule packets belonging to the same class of service on the link. Clearly, any policy that provides class isolation can be used to provide session isolation. If this is the choice, then FIFO is often used to schedule packets belonging to a single session. Such an approach suffers because no benefit is obtained (statistical multiplexing gain) from sharing the link between a group of sessions. To ameliorate this problem, several policies have been proposed for scheduling sessions belonging to the same class of service. These include FIFO, Earliest-due-date (EDD), FIFO+, and Jitter earliest due date (J-EDD). We will discuss each of these in turn later in this section.

Before beginning our description and discussion of class isolating policies and intra class scheduling policies, we briefly mention that policies can be further characterized as either idling or non-idling policies. Here a policy is said to be non-idling if it never allows the server to idle when there are packets queued at that server. A policy is said to be an idling policy if it is allowed to idle the server while there are packets queued for it. Although idling appears to be wasteful, it provides *predictability* which simplifies the problem of providing deterministic delay bounds as we will see shortly. Figure 3 provides a high level functional diagram of a link scheduler.

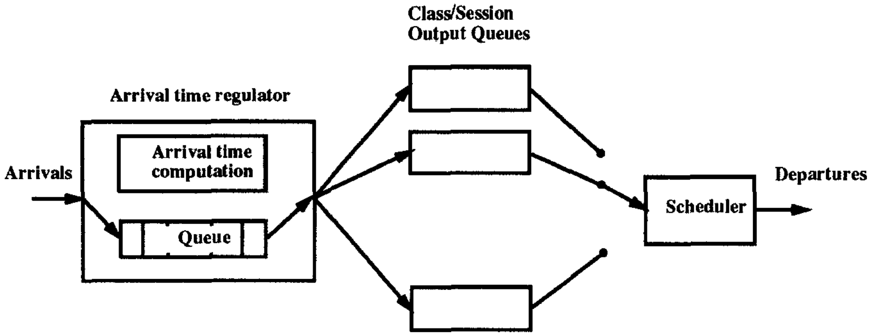

Figure 3: Link scheduler.

5.1 Class isolating scheduling policies

Hierarchical round robin (HRR): In its simplest form HRR is time division multiplexing. Time is divided into periods of constant length F and each session is given a fixed number of slots within that frame. Consequently, HRR is an idling policy. The maximum packet delay for a session can be bounded by $2F$ by choosing the fraction appropriately, provided that the source is described as a LBAP.

Sessions can be placeded into groups with each group being assigneda fraction of the frame. The groups can then handled using round robin and each group can have its own scheduler for scheduling packets belonging to that group. One choice fo a group scheduler is the round robin policy. This provides the basis for HRR in its most general form.

In its most general form, HRR consists of $N > 1$ levels. Associated with level i , $1 \leq i \leq N$, is a set of sessions, a frame size F_i, and parameter b_i. Consider the operation of level i. During the first b_i slots of that frame, the server is dedicated to the sessions associated with levels $i+1$ and above. During the remaining $F_i - b_i$ slots of the frame, the server is assigned to sessions belonging to level i. The packets belonging to levels $i+1$ and above are scheduled using HRR having levels $i+1, \ldots, N$. Thus multilevel HRR can be recursively defined in this way. Note that HRR operating at level N can provide all of the slots within its frame to the sessions assigned to that level.

Multilevel HRR provides suficient flexibility to provide different determinstic guarantees to different applications. Furthermore, due to the fact that a session may not use more than its share of slots within a frame, determinstic jitter guarantees can be made as well.

Although, as defined, HRR seems to be geared to applications with determinstic QOS requirements, in fact, best effort applications can be easily introduced by assigning it a priority lower than the determinstic class. In particular, best effort packets can use slots that would normally go idle.

Last, This policy is easy to implement. See [29] for further details on its implementation and performance.

Stop and Go Queueing (SG): This policy is similar to HRR with one important distinction. In addition to associating a frame to the outgoing link, a frame of the same length is also associated with each incoming link. These frames are then mapped to frames on the outgoing link by introducing a constant delay θ, where $0 \leq \theta < F$. This has the advantage that the end-to-end jitter can never exceed $2F$, no matter how long the path traversed by a session. The maximum delay on a link is $2F$ as under HRR.

As with HRR, it is possible to define a multiple level Stop-and-Go policy. Here F_i is assumed to be a multiple of F_{i+1}. Thus, an application having a delay bound of D at the node would be assigned to level $J = \arg\min\{i : 2F_{i-1} < D \leq 2F_i\}$. See [20] for further details and [19] for a description of how it can be integrated with other traffic classes

Weighted Fair Queuing: This policy was first introduced as a mechanism to ensure fairness between different sessions in traditional data networks[13]. It is most easily described under the assumption that workloads generated by sessions can be treated as infinitely divisible fluids. Let S sessions labelled $i = 1, \ldots, S$ share a link with capacity r. Associated with the sessions are parameters $\{\phi_i\}_{i=1}^{S}$ which determine the rates at which they receive service. Each session sees a link with capacity at least as large as $\phi_i / \sum_{j=1}^{S} \phi_j$. More precisely, if the set of sources with queued packets at time t is $\mathcal{S}(t) \subseteq \{1, \ldots, S\}$, then source $i \in \mathcal{S}(t)$ receives service at rate $\phi_i / \sum_{j \in \mathcal{S}(t)} \phi_j$. This policy ensures that, if a source has data to send during the period $[s,t]$, then the amount of its data transmitted is at least as large as $(t-s)\phi_i / \sum_{j=1}^{S} \phi_j$.

As described above, this policy is not implementable since the smallest unit of data transfer is a packet. However, it is not too difficult a task to accurately approximate the above policy by a dynamic priority policy. In particular, a packet is assigned a priority equal to the time that it would complete under WFQ. Let $\hat{T}_i$ an T_i denote the response times of the i-th packet under the preemptive WFQ and the non-preemptive version of WFQ respectively. If r denotes the rate of the server and L_{max} the maximum packet length, then it has been shown ([38]) that $|T_i - \hat{T}_i| \leq L_{max}/r$.

This policy was originally proposed and studied in [13] with $\phi_i = 1/S$ for the purposes of providing fair service to traditional data traffic in the internet. However, it has recently been shown that, if a source is a (σ, ρ) LBAP, it is possible to choose values for ϕ so that the maximum delay at the server is bounded [38]. In particular, if $\phi_i = \rho_i$, then

$$T_i \leq \frac{\sigma_i}{\rho_i} + L_{max}/r \tag{1}$$

It is unnecessary for other sessions to behave as LBAP's. Thus this policy can be used to share the server among applications requiring hard real time guarantees, statistical guarantees, and no guarantees. We will observe in the next section that this policy is the cornerstone of an approach proposed by Clark, et al. for providing QOS in an integrated digital services network.

5.2 Intra-class scheduling policies

The most commonly used policy in this group is FIFO. This is due to several reasons. First, it is extremely easy to implement. Second, it exhibits a number of important properties. For example, it is known to minimize the variance in the delay through a node and, in certain cases, the end-to-end delays through a tandem network [45]. It is also known to stochastically minimize the maximum end-to-end delay. These two features make it the policy of choice within a session and an option to consider for scheduling packets from different sessions belonging to the same class of service.

It is not clear that FIFO is an appropriate policy for scheduling packets belonging to either the deterministic or statistical service classes. Recently, a number of deadline based policies have been proposed. All of these are based on classical real-time scheduling policies such as rate monotonic (RM) and earliest due date (EDD).

Under EDD, each packet has associated with it a *due date*, typically interpreted as the time by which to complete transmission of the packet. At the time that the server becomes available, it is assigned the packet with the smallest (earliest) due date to transmit. Ferrari and Verma [15] propose this policy as part of a QOS provision mechanism where the due-date associated with a session is negotiated at the time that a call is admitted. It is also a component of the MARS approach [27].

A problem inherent in FIFO and EDD is that end-to-end jitter tends to grow as a function of the path length in networks. As a consequence, Verma and Ferrari [48] proposed an idling version of EDD, Jitter-EDD (J-EDD) which has the provable property that end to end jitter never exceeds that for a single node, regardless of path length. This property results from the addition of an input regulator (see Figure 3) which holds a packet until the time it is expected to arrive at the node. At that time it is released into the node to be scheduled.

More recently, Clarke, Shenker, and Zhang proposed a non-idling policy for controlling end-to-end jitter over long paths ([11]). Unlike J-EDD which holds a packet at a server until its expected arrival time, FIFO+ attempts to reduce jitter by giving higher priority to packets that have taken an inordinately long time in reaching the server and low priority to those that have arrived more quickly. Consider session s. Associated with server j_k on its path is a target delay $t_{s,k}$. Let $T_{s,k,j}$ be the actual delay incurred by the j-th packet from session s at server j_k. Then packet j is given a due-date of $\sum_{u=1}^{k-1} T_{s,u,j} - \sum_{u=1}^{k} t_{s,u}$ at server j_k. It is suggested that $t_{s,k}$ be set to the expected packet delay for session s and that it be updated as it changes. Very preliminary results indicate that it appreciably reduces the variance in the end-to-end response time over a policy such as FIFO. Last, it can be used in combination with other policies such as WFQ.

5.3 Inter class sharing policies

Minimum laxity threshold (MLT): The policies so far either partition the bandwidth between different classes of applications or even sessions (WFQ) or apply to a single application class. Recently, several policies have been proposed that attempt to deal with the QOS requirements of two or more distinct classes of applications in a complementary manner. One of the earliest such policies, Minimum Laxity with Threshold (MLT), introduced by Chipalkatti, et al. [10] deals with an application in which the QOS metric is the fraction of packets whose delay exceeds a deadline D and another application whose QOS metric is expected delay. They propose a policy that schedules the first class of packets whenever one or more of them are within d units of time from their deadline and serves the second class of packets otherwise. Their preliminary results show that a threshold can be chosen that tradeoffs the QOS of both classes thus yielding better performance than what mght be achieved by either a static priority scheme or FIFO.

More recently, Lazar, et al. [27], have proposed a more sophisticated variation of MLT that interleaves the transmission of deterministic and statistical packets in such a way that statistical packets are given priority so long as no deterministic packet is allowed to miss its deadline [27].

5.4 Buffer management policies

So far no mention has been made of the fact that buffer capacity at the link is finite in capacity, much less the impact that this has on link scheduling (if any). For the most part, the problem of buffer management is orthogonal to the subject of this paper. We assume that there are sufficient buffers for applications requiring deterministic QOS provided that other resources are available. In the case of applications requiring statistical guarantees, overflow is possible. The problem of choosing a packet discard policy has received less attention than it deserves. However, the proper choice must address two questions. Which session to choose from and, within a session, which packet to choose. The first question has not been satisfactorily answered. If there are best effort packets available, then discards should be made from them. If not, then there is the question of whether to spread discards over many statistical sessions or over a few.

6 Call Admission

The problem of how to do call admission is very complex and has generated a number of potential solutions. As described in section 3, a complete solution may require rate control/traffic shaping mechanisms, new link scheduling policies, routing policies and traffic monitoring mechanisms. Furthermore, call admission must also account for the different classes of traffic that are envisioned. These include minimally

1. deterministic,
2. statistical, and
3. best effort

as described in section 2. These may further subdivide into subclasses that differ from each other according to their associated QOS metrics.

Rather than attempt to provide complete solutions from the outset, we will focus on the problem of call admission for each class separately.

6.1 Deterministic guarantees

We begin with a discussion of how deterministic guarantees can be made for networks using session isolating schedulers such as HRR and SG. Consider the case where a deterministic session traverses a path Γ_s consisting of h hops. If the frame size at each link is F, then the end-to-end delay is bounded by $2hF$ under both HRR and SG. Multilevel versions of these policies can provide different delay bounds, one for each level. Thus deterministic applications can be divided into subclasses associated with the level that provides the appropriate delay bound. One problem with call admission based on HRR and SG is that it is not possible to decouple the delay bounds from bandwidth allocations. Applications requiring tight delay bounds will tend tobe allocated high bandwidths and vice versa. aathis is less true for HRR than for SG but is present nevertheless. Table 1 compares these two policies.

	Delay Bound	Jitter Bound
HRR	2hF	-
SG	2hF	2F

Table 1: Comparison between SG and HRR.

We turn our attention to call admission in the case that local schedulers use deadline based policies (EDD, J-EDD). Recall that these policies require that each session have a local deadline associated with it. Hence a responsibility of the connection set up phase is the choice of these local deadlines. The use of local deadlines also provides nodal isolation. Hence, checking on whether or not the establishment of a connection will affect other sessions need not proceed outside of the path in question.

Consider a new session, s, desiring to establish a connection on path Γ_s. During the first phase of call admission, each node determines whether or not there exists a local deadline which it can guarantee the delays of session s to fall below while maintaining the local guarantees for all other sessions. Examples of such calculations along with their computaional costs can be found in [15, 48]. Let d_k denote this minimum local deadline at node j_k. If all nodes on the path are able to calculate such deadlines, then the destination checks whether or not the end-to-end deadline can be met, i.e., is $\sum_{k=1}^{n_s} d_k \leq D_s$? If so, then the destination allocates local deadlines, D_k to the nodes on Γ_s sothat $d_k \leq D_k$ and $\sum_{k=1}^{n_s} D_k = D_s$. Several ways of allocating end-to-end deadline to local deadlines are described in [15]. This approach can produce provable guarantees using EDD and J-EDD as the local schedulers. The latter provides lower jitter guarantees than the former.

The following question arises: is it possible to provide guarantees under non-idling policies other than EDD? The answer is yes. In a very interesting series of papers, Cruz [12] shows that delays are bounded for a set of (possibly non-identical) sessions under *any set of non-idling policies* traversing a feed forward network provided that 1) each session (i) is described by a LBAP with parameters (σ_i, ρ_i) and for each link k, $r_k > \sum_{m \in \mathcal{S}_k} \rho_m$ where $\mathcal{S}_k$ is the set of sessions traversing link k. These results also apply to some non-feed-forward networks.

Unfortunately, the bounds are not useful for call admission. The bounds can be very very loose. For example, consider an h hop path consisting of T1 links shared by 48 32Kbs voice calls (see Table 2. The bounds appear to grow exponentially as a function of the path length. In addition, the bounds are not easy to compute and the results for non-feed-forward networks incomplete.

Following in the footsteps of Cruz, Parekh developed end-to-end delay bounds for a session s described by an LBAP in an arbitrary network operating under

h	WFQ	FIFO	SG
1	16	16	48
2	32	48	80
3	49	115	113
4	65	264	145
5	81	623	177
6	98	1,572	210
7	114	4,382	242
8	130	13,835	274
9	147	50,676	307
10	163	220,010	339

Table 2: Comparison of bounds for FIFO, SG, and WFQ.

WFQ under very reasonable assumptions. The theory exhibits the folllowing important properties.

- Networks can be non-feed-forward,
- In the case that $\phi_k = \rho_s$ for $j_k \in \Gamma_s$, the delay bound is given by

$$T \leq \frac{\sigma_s}{\rho_s} + \text{propagation delay} \tag{2}$$

  which is independent of the path length.

Clarke, et al., [11] have proposed WFQ as the scheduler in their architecture for providing deterministic QOS. The quality of the bounds provided by WFQ can be found in Table 2 and can be compared with those of FIFO and SG for our voice example.

6.2 Statistical guarantees

Considerable work has been conducted on the design and evaluation of call admission policies for sessions requiring one or the other of the following two guarantees,

$$\begin{aligned} \Pr[T \leq D] &< \epsilon \\ \Pr[\text{packet loss}] &< \epsilon. \end{aligned}$$

We will consider each of these in turn.

6.2.1 Statistical deadline guarantees

Consider a session that requires a guarantee on the tail of the end-to-end delay distribution. One approach to handling this session is to treat it as if it has the following deterministic QOS requirement, $T \leq D$. However, there is considerable evidence that such an approach will result in extremely poor performance, i.e., the number of sessions permitted to use the network will be considerably lower than necessary. We illustrate this with an example taken from [51].

Figure 4 illustrates a network (labelled M4) of 3 × 3 switches connected by T1 rate lines. The network has been configured so that each communication link carries 48 32Kbs voice calls where each voice call generates a packet every 16ms during a talkspurt. Talkspurt lengths are assumed to be exponentially distributed random variables with mean 352 ms and silence periods are assumed to be exponentially distributed rv's with mean 6.5 ms. (Silence periods are typically much larger. This mean was chosen so that the link utilizations would be all approximately 98%.)

Figure 5 shows simulation results for the distribution of the end-to-end delay of a session traversing links S1 - S4. The estimate for the distribution is taken from ten independent simulation runs, each of which simulated approximately 1/2 million packets for those sessions whose path length was four. Also shown are the bounds obtained for WFQ, SG, and FCFS. The results illustrate how poorly the bounds on maximum delay can be for the tail of the end-to-end delay distribution.

Consequently, a different approach is required for performing call admission when applications require statistical guarantees. In this section we describe several such approaches and discuss their advantages and disadvantages. These include

- provable guarantees,
- and approximate guarantees.

The first approach is an extension of the work of Cruz for bounding maximum end-to-end delay to bounding the tail of the end-to-end delay distribution. Kurose [33] provides tail bounds for the same network as Cruz under the assumption that busy periods at each node are finite and bounded in duration. Inherent in the model is the asumption that sources are described by LBAP's. Yaron and Sidi [50] and Chang [9] consider more general arrival processes for which the peak rate is unbounded and, based on Chernoff's bound, develop bounds on the end-to-end delay distribution. Although these models can be used to provide tighter bounds than the model of Cruz, preliminary evidence indicates that they remain still too loose to be of practical use in call admission. Furthermore, they share some of the problems inherent in Cruz's model, high computational complexity, not suitable for most general networks, etc...

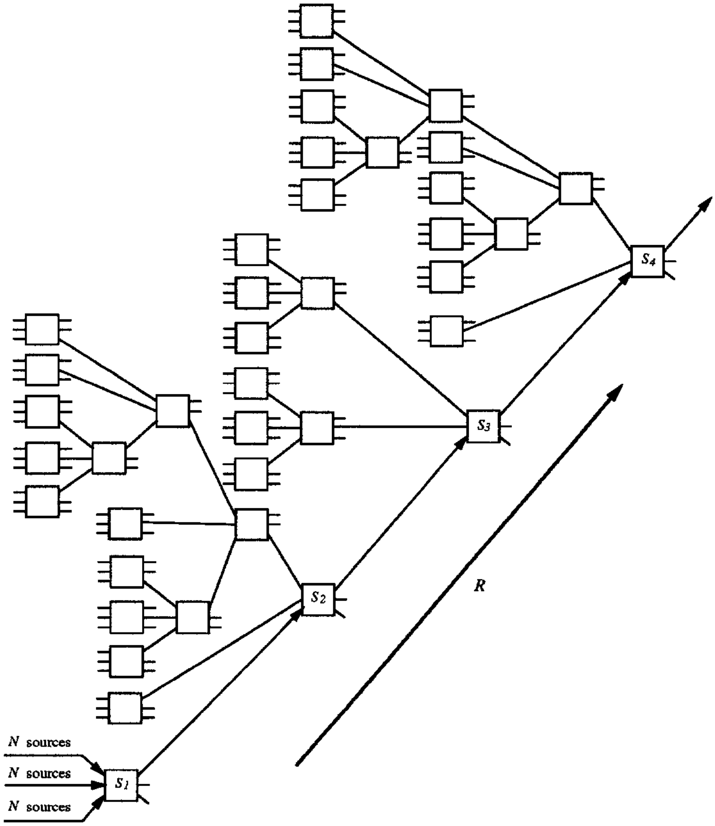

Figure 4: M4 Network.

This brings us to the approach most prevalent in the literature, namely to develop call admission policies that *approximately* provide statistical guarantees. Two approaches have been taken in this direction. The first is to develop heuristic models for estimating the tails of delay distributions [15, 28]. The latter approach actually measures current network behavior and uses this to parameterize the model. The second is to classify applications into different classes and perform analysis or simulations off-line to determine the number of statistical sessions that can be admitted and, more generally, the number of statisical sessions that can be combined with different numbers of deterministic sessions. This is exemplified in the work described in [27].

6.2.2 Statistical loss guarantees

We begin the discussion of how to provide statistical loss guarantees by first noting that a solution to this problem often automatically solves the problem of

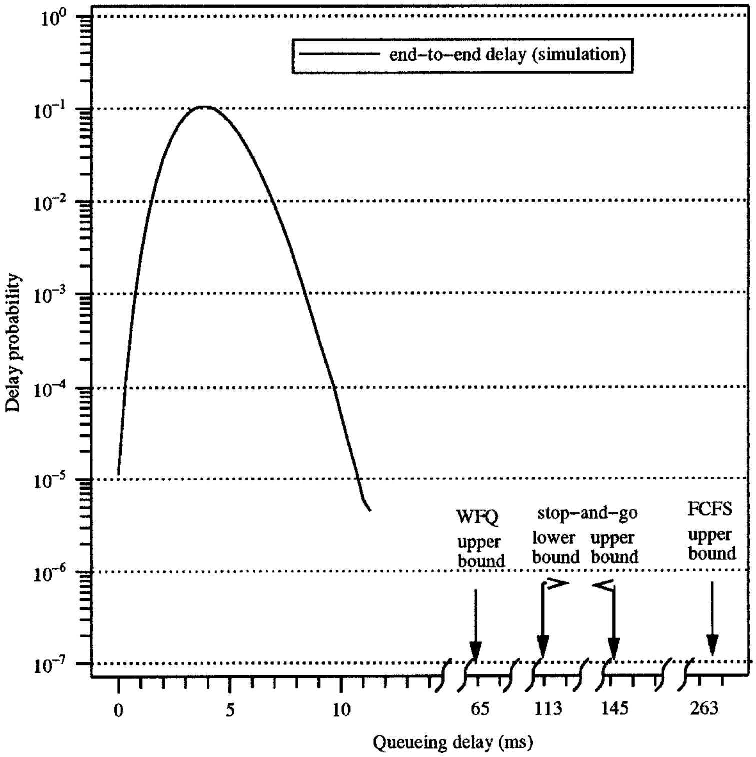

Figure 5: Delay comparison for M4 network.

providing statistical deadline guarantees in many high speed networks. Consider a network with links having bandwidths of 150Mbs. Let the scheduler at each link be FIFO and let the buffer capacity be 800Kb (approximately 100 1Kbyte packets). The delay through a 5 hop path is bounded by 30ms (excluding propagation delay) which is tolerable for real-time applications such as voice and video. Hence, the event $T > D$ corresponds to the event of a packet that is lost due to buffer overflow in this case.

Unlike deadlines, there is no general method for obtaining provable bounds on packet loss probabilities except in the case of a single link. Saito [42] provides bounds on cell loss probabilities for an ATM switch using a FIFO scheduler. The bounds require the average rate and the peak rate for each source over an interval of length $K/2$ where K is the buffer capacity. As the model does not derive expressions of this peak rate at the output of the switch, it is not able to handle more than one multiplexer.

Once we relax our requirement that the guarantee be provable, then we find

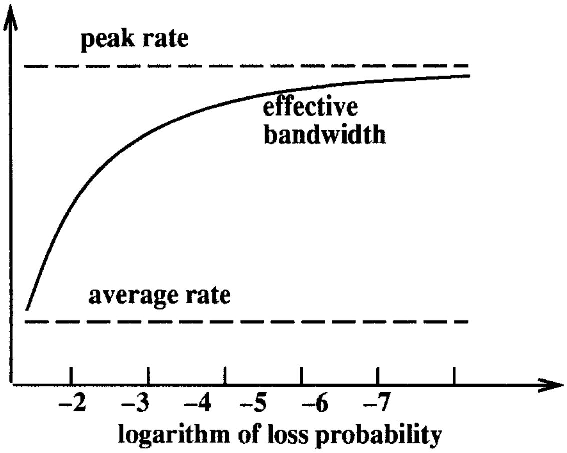

Figure 6: Effective bandwidth as a function of QOS requirement.

that the primary approach to providing statistical loss guarantees is based on the *theory of effective bandwidths.*

The theory of effective bandwidths originated in the context of a single link. Consider a single session, modelled as an on-off source with peak rate R_{peak}, mean burst length b, and utilization (fraction of time in the on state) u feeding a buffer with capacity K. Assume that it has a loss requirement Pr[packet loss] $\leq \epsilon$. The effective bandwidth $\hat{c}$ is the service rate required to serve this source so that its QOS requirement is met. The effective bandwidth as a function of the QOS requirement is illustrated in Figure 6. It falls between the peak and average rates and increases as the QOS requirement becomes more stringent.

Assume for now that the effective bandwidth of a source is easily calculated and that the effective bandwidth corresponding to a set of sources can be expressed as the sum of their individual effective bandwidths, i.e., $\sum_{i \in S} \hat{c}_i$. Then, the problem of call admission is simplified tremendously. The decision to accept a new call s requires the following test, is $r - \sum_{i \in S} \hat{c}_i > \hat{c}_s$? This approach was first proposed in [21] where the following heuristic expression was given to compute the effective bandwidth of an on-off source,

$$\hat{c} = R_{peak} - \frac{x - \sqrt{[\alpha b(1-u)R_{peak} - K]^2 + 4K\alpha bu(1-u)R_{peak}}}{2\alpha b(1-u)} \tag{3}$$

where $\alpha = \ln(1/\epsilon)$. This expression was derived from a fluid model of an off-on source, [3], and developed to be 1) simple to compute, depending on only three parameters and 2) generally pessimistic.

One problem with this approach is that it ignores the possible multiplexing gains achieved by sharing the link among a number of sources. Hence, the following expression in [21] was proposed for the effective bandwidth of a collection of sources S,

$$\hat{C} = \min\left\{ m + \alpha'\sigma, \sum_{i\in\mathcal{S}} \hat{c}_i \right\} \tag{4}$$

where m is the average aggregate rate of the sources in $\mathcal{S}$, σ is the standard deviation in the aggregate rate, and $\alpha' = \sqrt{-2\ln\epsilon - \ln(2\pi)}$. The first expression in the min is based on a Gaussian approximation of the aggregate bit rate. Such approximations have been shown to accurately model the stationary bit rate when the number of sources is sufficiently large (¿10 is suggested in [22]).

This has been extended to a network setting by assuming that the source traffic characteristics are relatively unaffected much when passing through a link (see [22] for evidence for the validity of this assumption) and through an application of the principle of nodal isolation, i.e., allocating the end-to-end packet loss requirement among the nodes on a path. See [22] for details of this approach.

There is currently insufficient experience with this approach to determine how well it will work. As mentioned earlier, the guarantees are approximate. In order to compensate for this, they have been chosen to be very conservative. However it is not known how conservative they are. Further work is required to evaluate this approach.

There have been several other proposals on how to calculate effective bandwidth for the case of identical sources sharing identical loss requirements. They are based on the following approach - for a given loss requirement, determine the maximum number of sources, n_{max}, that can share a link so that they all satisfy their requirements. Then $\hat{c} = r/n_{max}$. This can be obtained either through analysis, [1], or simulation, [16].

Observe that the test for call admission can be stated in the following equivalent form, is $n + 1 \leq n_{max}$? Here n denotes the number of sessions currently using the link. Two types of sessions can be handled in this way by calculating a feasible region (see Figure 7) which indicates the different combinations of sessions of the two types that can be accommodated while satisfying the loss requirements of both. Again, a call can be admitted if the combined populations fall within the feasible region. Such an approach has been sggested and studied in [25]. in the context of loss requirements and in the context of different types of QOS requirements [27].

Last, the work of Guerin, et al., which first proposed and developed the theory of effective bandwidths through simple heuristic arguments has been placed on a solid mathematical basis by several recent papers, [17, 14, 31]. This work has resulted in two types of results. The first is the derivation of the effective bandwidth of a single source for a large class of queueing systems and traffic models. Typically these relate to the dominant eigenvalue of the rate matrix associated with a Markovian model of the source. Second, it has been shown in

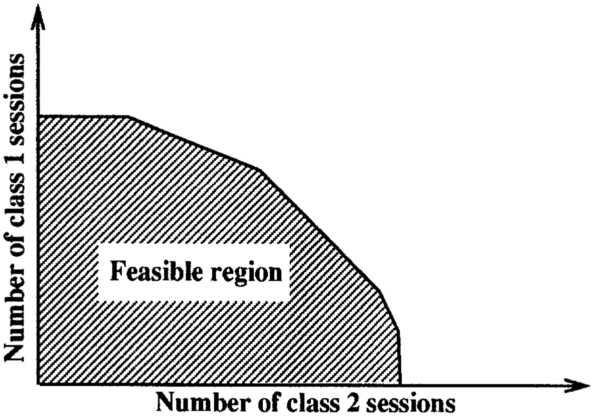

Figure 7: Feasible region for call admission.

the limit as $K \to \infty$ and $\epsilon \to 0$ that the effective bandwidth corresponding to a collection of sessions is equal to the sum of the effective bandwidths asociated with the sources for Markovian arrival processes. Hence, the assumption that the effective bandwidth of a collection of sessions can be approximated by the sum of the effective bandwidths of the individual sources is quite reasonable for low loss probabilities.

6.3 Best effort

This class of applications is easily handled. In a real implementation it may be useful to allocate a fraction of the bandwidth to best effort traffic. This is easily done with most of the approaches that we have described as they either are based on the principle of session isolation or they require knowledge of peak rates.

6.4 Other issues

There are a number of issues and problems that have not been adequately addressed, either in this section or in the literature. These ibclude

- allocation of end-to-end QOS requirements to nodal QOS requirement,
- adaptivity to changes in link loads, and
- adaptivity to changes in session characteristics.

Although several proposed call admission algorithms require the allocation of end-to-end QOS requirements to nodal requirements, e.g., [15, 22], the problem has not been thoroughly addressed. One exception is the work of Nagarajan [36] which compares an optimal allocation to the simple heuristic of equal allocation. This study shows that, if the QOS requirement is low loss, then little is gained in using the optimal allocation over the equal allocation. On the other hand, is the QOS requirement is a mean delay bound, then an optimal allocation can provide significantly better performance than an equal allocation. A sensitivity measure, the relative gain ratio, is proposed which can be used to perform a quick test to determine whether it is useful to try to find the optimal allocation.

A second issue that is only now beginning to receive attention is that of changes in network loads, source characteristics, etc.. Is the contract negotiated between network and user static during the life of the session or can it be renegotiated? For example, if a user declares itself as being characterized by a high value of R_{peak} but the network determines through measurements that it is considerably lower, can the network take advantage of it? Several proposed algorithms allow the network to modify network resource allocations to sessions in order to account for changes in traffic characteristics, see [2].

7 Summary and Future Work

In this paper we have presented the state of the art of provision of QOS in integrated digital services networks. We have focussed primarily on the problem of call admission, and specifically the question *can a new call be admitted with the QOS that it desires while maintaining the QOS of all sessions presently in the network?* We have seen how this is intricately related to the choice of scheduling policy at each link. We have described a number of approaches to solving the problem of call admission based on the principles of nodal and session isolation as well as approaches that attempt to share resources between sessions of different classes. At this point in time there is has been very little comparison between different approaches - either from the point of view of assumptions regarding the underlying network architecture or from the point of view of performance. Much remains to be done in this area.

One interesting dichotomy exists in the different approaches reported on how to provide QOS that has yet to be dealt with satisfactorily. This is the division between the approaches that deal primarily with delay and those that ignore delay but deal with buffer overflow. It seems clear that the current real-time applications such as voice and video can be handled in a satisfactory manner by the latter approach provided that raw bandwidth is at least 100Mbs. It remains to be seen whether applications will be developed which will require deadline constraints that are only slightly larger than propagation delays or whether there will be a large number of low speed networks for which the algorithms providing delay guarantees will be required. This is an area worth investigating.

Another area worth investigating is the application of some of the approximate guarantee techniques developed for delays [28] to the problem of packet loss. Another fruitful area of research is that of either developing call admission algorithms based on interval QOS metrics or trelating such metrics more closely to stationary metrics such as Q2 and Q4 so as to use existing approaches for dealing with interval metrics.

Last, the development of protocols for dealing with call admission or the data transport is in its infancy. Furthe work is required in this area.

Acknowledgments: I would like to acknowledge Ramesh Nagarajan for the endless discussions regarding the behavior and underlying assumptions of different scheduling algorithms and call admission algorithms. I would also like to thank David Yates for supplying me with the numerical and simulation results comparing different call admission policies.

References

[1] S. Akhtar. Congestion control in a fast packet switching network. Master's thesis, Washington University, St. Louis, Missouri, 1987.

[2] J.-T. Amenyo, A.A. Lazar, and G. Pacific. Cooperative distributed scheduling for ats-based broadband networks. In *INFOCOM'92*, pages 333–342, May 1992.

[3] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61:1871–1894, 1982.

[4] Caglan M. Aras, Jim Kurose, Douglas Reeves, and Henning Schulzrinne. Real-time communication in packet-switched networks. To Appear in Proceedings of the IEEE, January 1994.

[5] Ernst Biersack. Error recovery in high-speed networks. In *Second International Workshop on Network and Operating System Support for Digital Audio and Video*, page 222, 1991.

[6] Hugh S. Bradlow. Performance measures for real-time continuous bit-stream oriented services: Application to packet reassembly. *Computer Networks and ISDN systems*, 20:15–26, 1990.

[7] CCITT. Rec. I.121: Recommendation on broadband aspects of ISDN. 1988.

[8] CCITT. Rec. I.371: Recommendation on traffic control and congestion control in B-ISDN. 1992.

[9] C.-S. Chang. Stability, queue length and delay, part ii:stochastic queueing networks. Technical Report RC 17709, IBM.

[10] R. Chipalkatti, J.F. Kurose, and D. Towsley. Scheduling policies for real-time and non-real-time traffic in a statistical multiplexer. In *IEEE INFOCOM'89*, pages 774–783, April 1989.

[11] D.D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: architecture and mechanism. In *ACM SIGCOMM'92*, pages 14–26, 1992.

[12] Rene Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37:132–141, 1991.

[13] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internetworking:Research and Experience*, 1:3–26.

[14] Anwar Elwalid and Debasis Mitra. Effective bandwidth of general Markovian traffic sources and admission control of high-speed networks. Submitted to ACM-IEEE Transactions on Networking, July 1992.

[15] Domenico Ferrari and Dinesh Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE J.Select.Areas Commun.*, 8:368–379, April 1990.

[16] G. Gallassi, G. Rigolio, and L. Fratta. ATM:bandwidth assignment and bandwidth enforcement policies. In *GLOBECOM'89*, pages 1788–1793, December 1989.

[17] R. J. Gibbens and P. J. Hunt. Effective bandwidths for multi-type uas channel. *QUESTA*, 9:17–28, 1991.

[18] André Girard. *Routing and Dimensioning in Circuit-Switched Networks.* Addison Wesley, 1990.

[19] S. J. Golestani. A stop-and-go queueing framework for congestion management. In *Proc. 1990 SIGCOMM*, pages 8–18.

[20] S. J. Golestani. Congestion-free transmission of real-time traffic in packet networks. In *IEEE INFOCOM'90*, pages 527–536, June 1990.

[21] Roch Guerin et al. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE J.Select.Areas Commun.*, 9(7):968–981, 1991.

[22] Roch Guerin and Levent Gun. A unified approach to bandwidth allocation and access control in fast packet-switched networks. In *INFOCOM'92*, pages 01–12, 1992.

[23] S. Keshav H. Zhang. Comparison of rate-based service disciplines. In *SIGCOMM.*

[24] Harry Heffes and David Lucantoni. A markov modulated characterization of voice and data traffic and related statistical multiplexer performance. *IEEE J.Select.Areas Commun.*, SAC-4:856–867, September 1986.

[25] Joseph Y. Hui. Resource allocation for broadband networks. *IEEE J.Select.Areas Commun.*, 6(9):1598–1608, December 1988.

[26] Ren-Hung Hwang. *Routing in high-speed networks.* PhD thesis, University of Massachusetts, Amherst, 1993.

[27] Jay M. Hyman et al. Real-time scheduling with quality of service constraints. *IEEE J.Select.Areas Commun.*, 9(7):1052–1063, September 1991.

[28] S. Jamin, S. Shenker, L. Zhang, and D.D Clark. An admission control algorithm for predictive real-time service(extended abstract). In *Third International Workshop on network and operating system support for digital audio and video*, pages 308–315, November 1992.

[29] C.R. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks. In *Globecom'90*, December 1990.

[30] Gunnar Karlsson and Martin Vetterli. Packet video and its integration into the network architecture. *IEEE J.Select.Areas Commun.*, 7(5):739–751, June 1989.

[31] F. P. Kelly. Effective bandwidths at multi-class queues. *QUESTA*, 9:5–16, 1991.

[32] S. Keshav. On the efficient implementation of fair queueing. *Internetworking:Research and Experience*, 2:157–173.

[33] Jim Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *ACM SIGMETRICS'92*, pages 128–139, June 1992.

[34] Z. Liu and D. Towsley. Burst reduction properties of the token bank in ATM networks. In *IFIP workshop on modeling and performance evaluation of ATM technology.*

[35] Zhen Liu and Don Towsley. Burst reduction properties of rate-control throttles:departure process. To Appear in *Annals of Operations Research.*

[36] Ramesh Nagarajan. *Quality-of-service issues in high-speed networks.* PhD thesis, University of Massachusetts, Amherst, 1993.

[37] Pramod Pancha and Magda El Zarki. Bandwidth requirements of variable bit rate MPEG sources in ATM networks. In *IFIP workshop on modeling and performance evaluation of ATM technology*, pages 5.2.1–5.2.25, January 1993.

[38] Abhay Parekh and Robert Gallager. A generalized processor sharing approach to flow control in integrated services networks - the single node case. In *INFOCOM'92*, pages 915–924, 1992.

[39] V. Ramaswamy. Traffic performance modeling for packet communication whence, where and wither. In *Third Australian Teletraffic Seminar*, November 1988. Keynote Address.

[40] V. Ramaswamy and Walter Willinger. Efficient traffic performance strategies for packet multiplexers. *Computer Networks and ISDN systems*, 20:401–407, 1990.

[41] J.-F. Ren, J.W. Mark, and J.W. Wong. Performance analysis of a leaky-bucket controlled ATM multiplexer. To Appear in *Performance Evaluation.*

[42] H. Saito. Call admission control in an ATM network using upper bound of cell loss probability. *IEEE Transactions on Communications*, 40(9):1512–1521, September 1992.

[43] Nachum Shacham. Packet recovery in high-speed networks using coding and buffer management. In *INFOCOM*, pages 124–131, 1990.

[44] M. Sidi, W. Liu, I.Cidon, and I. Gopal. Congestion control through input rate regulation. In *GLOBECOM'89.*

[45] D. Towsley and F. Baccelli. Comparisons of service disciplines in a tandem queueing network with real-time constraints. *OR Letters*, 10.

[46] J. Turner. New directions in communications (or which way to the information age). *IEEE Communications Magazine*, 24:8–15, 1986.

[47] J.S. Turner. Managing bandwidth in ATM networks with bursty traffic. *IEEE Network*, 6(5):50–59, September 1992.

[48] Dinesh Verma, Hui Zhang, and Domenico Ferrari. Delay jitter control for real-time communication in a packet switching network. In *IEEE Tricomm'91*, April 1991.

[49] G.M. Woodruff and R. Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE J.Select.Areas Commun.*, 8.

[50] Opher Yaron and Moshe Sidi. Calculating performance bounds in communication networks. In *IEEE INFOCOM'93*, pages 539–546, April 1993.

[51] David Yates et al. On per-session end-to-end delay and the call admission problem for real-time applications with qos requirements. To appear in SIGCOMM'93.