Real-Time Communication in Packet-Switched Networks

Çağlan M. Aras,[1] James F. Kurose,[2] Douglas S. Reeves[3] and Henning Schulzrinne[4]

Abstract

The dramatically increased bandwidths and processing capabilities of future high-speed networks make possible many distributed *real-time* applications, such as sensor-based applications and multimedia services. Since these applications will have traffic characteristics and performance requirements that differ dramatically from those of current data-oriented applications, new communication network architectures and protocols will be required. In this paper we discuss the performance requirements and traffic characteristics of various real-time applications, survey recent developments in the areas of network architecture and protocols for supporting real-time services, and develop frameworks in which these, and future, research efforts can be considered.

[1]Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC 27695. This research was partially supported by a Graduate Fellowship from the IBM Corporation.

[2]Department of Computer Science, University of Massachusetts, Amherst MA 01003. The research of this author is supported in part by the National Science Foundation, under grant NCR-9116183, the Defense Advanced Projects Research Agency under contract NAG2-595, and the Motorola Codex Coroporation.

[3]Departments of Computer Science and Electrical and Computer Engineering, North Carolina State University. The research of this author has been supported by the National Science Foundation under grant CCR-9010771 and the Air Force Office of Scientific Research under contract F49620-92-J-0441.

[4]AT&T Bell Laboratories, Murray Hill, NJ.

1 Introduction

Computer networks are in a period of transition, moving from relatively slow communication links and data-oriented services to high-speed fiber optic links and a diverse set of services. Many of these services, such as voice, video and other applications, will have stringent *real-time* constraints and will demand not only high-bandwidths, but a predictable "quality of service" (QOS) not offered by current best-effort-delivery networks. The large amounts of bandwidth promised by future high-speed networks also offer the possibility of integrating such real-time applications together with more traditional data-oriented services within a single common network. Thus, while the scaling of bandwidth to more than a gigabit per second in next generation networks will certainly have a profound effect on all aspects of networking, the need to support a more diverse mix of services by accommodating the performance requirements of real-time applications raises important issues that go beyond bandwidth and bandwidth-delay product scaling.

Traditional communication network applications such as file transfer, electronic mail and remote login are examples of non-real-time applications, for which the performance metrics of interest are typically average message/packet delay and throughput. These applications also have strict reliability requirements; indeed, much of the complexity of traditional network protocols arises from the need for loss-free communication between data-oriented, non-real-time applications.

The characteristics of real-time communication applications differ significantly from those that are non-real-time. As in real-time computing, the distinguishing feature of real-time communication is the fact that the value of the communication depends upon the times at which messages are successfully delivered to the recipient. Typically, the desired delivery time for each message across the network[5] is bounded by a specific maximum delay or latency, resulting in a *deadline* being associated with each message. This delay bound is an application-layer, end-to-end timing constraint. If a message arrives at the destination after its deadline has expired, its value to the end application may be greatly reduced. In some circumstances messages are considered "perishable," that is, are useless to the application if delayed beyond the deadline. These messages are discarded and considered lost. For data-oriented applications, achieving low latency is usually desirable. However, some real-time applications do not care how much prior to a deadline a message arrives. Indeed, early arrival may even be considered harmful as it requires buffering at the receiver to achieve constant end-to-end delay.

Real-time communication applications are commonly classified as either soft or hard real-time. *Soft real-time* applications can tolerate some amount of lost messages, while *hard real-time* applications have zero loss tolerance. As we will see, the networking mechanisms required to handle traffic for these two kinds of applications can differ significantly. In general, soft real-time applications require less stringent service and thus allow the network to maximize network utilization. In hard real-time applications, deterministic predictability of network delays takes precedence over network utilization considerations.

[5] Generally only the queueing delay is discussed in this and other papers, as the packetization, switching and propagation delays are assumed known and fixed.

Another important performance metric for real-time traffic is *delay jitter*, commonly defined as the maximum variation in delay experienced by packets in a single connection. [6] For example, if the minimum end-to-end delay seen by any packet in a connection is 1 ms and the maximum delay is 6 ms, then the delay jitter of the connection is 5 ms. Many real-time applications, particularly those which are interactive, require a bound on jitter, in addition to a bound on the delay. As we will see, some methods of real-time communication specifically manage the jitter, while others do not. Note that certain applications such as non-interactive television and audio broadcasting, require bounds on jitter but not delay.[7]

The different performance metrics and reliability requirement of real-time traffic suggest that network protocols and architectures previously developed for data-oriented communication applications may not be well-suited for supporting real-time and integrated real-time/non-real-time applications.

1.1 Unsatisfactory Approaches to Real-Time Communication

There are several mechanisms for supporting real-time communication which we consider to be unsatisfactory. Circuit-switching, for instance, can provide real-time delivery guarantees very easily. A circuit-switched network simply sets aside a fixed portion of the network bandwidth according to the estimated peak bandwidth requirement of each application. As will be discussed later, real-time traffic is often bursty, leading to low effective bandwidth utilization unless idle time can be filled by non-real-time traffic. In addition, the typically coarse granularity of bandwidth allocation can lead to inefficiencies for the wide range of services expected to be carried in the integrated networks of the future [69].

Buffers at the receiver can be used to control jitter. The amount of buffer space required can be determined from the peak rate and delay jitter of the delivery process and can be quite large for a network with no control of delay. For example, a single video source transmitting 30 frames per second, each containing 2 Mb of data and experiencing a transmission jitter of 1 second, would require 60 Mb of buffer space at the destination to eliminate the jitter. On the other hand, the network can bound end-to-end jitter only by delaying packets, which requires storage within the network. The trade-off between shared high-speed memory within the network versus lower-speed dedicated memory at the receiver needs to be considered. In contrast to jitter guarantees, delay guarantees cannot be provided by buffering alone, as buffering can only delay the time of delivery.

It has been suggested that advances in transmission facilities will make bandwidth "too cheap to meter," so that low utilization can all but guarantee low delays for real-time services without

[6] We use the term "packet" to denote the entity of interest for scheduling, and as the object of performance guarantees. A packet may, for example, consist of several cells, depending on the underlying technology. A "connection" is a real-time communication session established between end-user applications at different sites. A connection is also sometimes called a stream, a call, or a channel.

[7] That is, the minimum delay can assume any value. This requirement is motivated by the desire to limit the size of the delay-smoothing buffer at the receiver.

special control efforts.[8] Three trends argue against this. First, end systems producing traffic have decreased their cost-to-speed ratio much more rapidly than transmission facilities. Secondly, new applications have tended to fill increased affordable bandwidth. And thirdly, low-bandwidth communication systems such as cellular radio are interesting targets for packetized communication to facilitate service integration.

A sounder argument may be made that even if utilization for *real-time* services is kept low, lower-priority data traffic can fill the gaps left by peak bandwidth allocation. At least in the initial stages of deploying integrated high-speed networks, data traffic originating on LANs is likely to dwarf traffic with real-time needs. Note that real-time traffic will likely produce more revenue per bit; this motivates the service provider to support high real-time utilization.

1.2 Goals for Real-time Communication Techniques

All methods of real-time communication aim to provide real-time message delivery with either low or zero loss rates (soft or hard real-time, respectively). The following are some desirable properties for real-time communication:

- low jitter
- low latency
- ability to easily integrate non-real-time and real-time services
- adaptable to dynamically changing network and traffic conditions
- good performance for large networks and large numbers of connections
- modest buffer requirements within the network
- high effective bandwidth utilization
- low overhead in header bits per packet or cell
- low processing overhead per packet within the network and at the end system

This paper aims to survey research on the new network architectures and protocols needed to support real-time services in packet-switched networks. Our focus is on wide-area networks, although many of the ideas discussed are equally applicable to local area networks. Occassionally, special mention is made to ATM [49], as it is the likely technology for carrying real-time packetized traffic.

The remainder of this paper is structured as follows. In the next section, we look at the characteristics of some of the applications that require real-time network services. Methods of hard real-time communication are discussed in section 3, while techniques for soft real-time communication are discussed in section 4. Section 5 concludes the paper with a list of some important open problems.

[8] The transport of audio and video within the current Internet operate on this basis.

2 Characteristics of Real-Time Traffic

A wide range of possible real-time communication applications are expected to co-exist in an integrated network. A partial list includes: multimedia conferencing [42], shared workspaces, remote medical diagnosis, telephony, command and control systems [10], distributed interactive simulation, audio and video broadcasts, and games.

Many of the traffic sources for which real-time service is desirable share characteristics that set them apart from traditional data traffic. In this section, we first focus on the general properties of data rate, packet size and loss tolerance; we then summarize work on characterizing the properties of particular sources of real-time traffic. During stream admission, these properties assist the network in determining the resources to be allocated to a particular real-time session. This characterization must be unambiguous, easy to specify, enforceable, and usable for reserving resources [36]. The traffic characteristics must be enforced both to 1) protect other applications from the effects of a misbehaving client, and 2) distinguish between negotiated traffic, which should continue receiving guaranteed service, and excess traffic, which may not.

Some real-time sources have inherent characteristics that distinguish them from typical data sources. For example, voice packets tend to be small to minimize packetization delays [83] and to limit the effect of packet losses [82]. The 48-byte cell size for ATM [116], for example, was chosen primarily for the benefit of voice applications – in particular, to avoid the use of echo cancellation equipment on continental connections. Also, small packets limit the amount of time a single packet can occupy the channel.

In order to predict the performance of communication systems carrying real-time data such as audio or video, an accurate source model has to be found. This is made difficult by the fact that the statistics of the traffic entering the network depend on the nature of the source material, the encoding method used, and the timing of packets by the encoder (a large packet every video frame, smaller packets equally spaced over the frame duration, or smaller packets transmitted at peak rate [102]). Thus, models for differing timescales may be needed [56, 90].

The description of sources is made easier by the fact that in many real-time applications, the source of the data is a sensor which samples a physical quantity to produce a digital signal. The sensor samples the physical quantity at regular intervals called the period T, and the data generated by the sensor is fed into the network as a real-time stream. Many such sources can be approximated by one of the following three source models, as shown in Fig. 1:

constant bit rate (CBR): Fixed-size packets arrive at deterministic intervals. Certain real-time applications, such as air-traffic control, generate data which has few redundancies and which is too important to be compressed in a lossy way. The data is generated by sensors at regular intervals.

variable bit rate (VBR):

- **on/off sources:** The source alternates between a period in which fixed-size packets arrive with deterministic spacing and an idle period. An example is voice traffic, discussed in more detail in section 4.1.

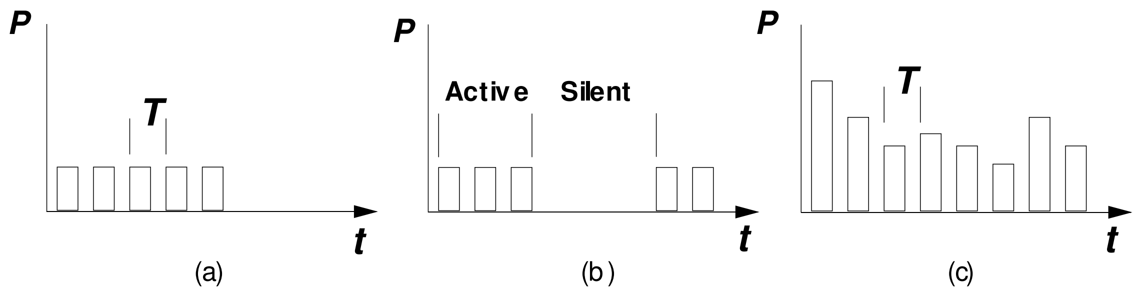

Figure 1: Packet arrivals from (a) continuous data sources, (b) voice sources with silence interval detection, and (c) compressed video sources(P=packet size, t=time, T=Packet interarrival time)

periodic with variable packet sizes: Each period, the source submits a single packet of variable length to the network. An example is video; different frames may experience varying compression ratios for the same output quality level. See section 4.1 for more discussion.

3 Hard Real-Time Communication

3.1 General Remarks

Some real-time applications require a guaranteed maximum delay and cannot tolerate any packet loss. As an example, consider a distributed process control system. In such a system, a message which indicates a reactor vessel is about to exceed its pressure limits must be received in time. Likewise, a response message which indicates the appropriate safety measures to take must be guaranteed a successful and timely delivery. A lost or late message in either case could be catastrophic. Hard real-time applications are thus intolerant of packet loss. The methods described in this section are intended to prevent losses due to buffer overflow and missed deadlines.

We distinguish between two classes of methods which provide hard real-time service in networks: the *rate-based* methods and the *scheduler-based* methods. For rate-based methods, the quality of service requested by a connection is translated into a transmission rate or bandwidth. There are a predefined set of allowable rates, which are assigned static priorities. The allocated bandwidth guarantees a fixed maximum delay for each packet in that rate class. The scheduler-based methods instead analyze the potential interactions between packets of different connections, and determine if there is any possibility of a deadline being missed. Priorities are assigned dynamically based on deadlines. Rate-based methods have the advantage of simple implementation, while scheduler-based methods allow bandwidth, delay, and jitter to be independently allocated.

In this section we discuss both classes of hard real-time communication methods. Section 4 will describe methods of soft real-time communication. The reader is also referred to a survey paper on hard real-time communication by Zhang and Keshav [130].

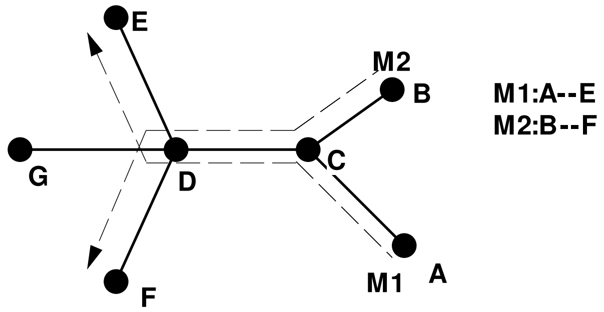

Figure 2: Example Network

In our communication model, the network is composed of a set of *nodes*,[9] connected by a set of *links*. Each unidirectional link ℓ^j allows two nodes to communicate with bandwidth C_j. The set of links that a packet of a connection i traverses in going from its source to its destination is called the *path* of the packet, denoted Γ_i. H_i is the number of hops on a path; Ω_j is the set of connections which use a link ℓ^j. The example in Figure 2 has two connections: connection M_1, following path $\Gamma_1 = \{AC, CD, DE\}$; and connection M_2, following path $\Gamma_2 = \{BC, CD, DF\}$. For link CD, $\Omega^{CD} = \{M_1, M_2\}$.

3.2 Real-Time Scheduling Theory

The theory of real-time scheduling has been developed and applied primarily to scheduling of jobs on a single processor [119]. For real-time communication, the link replaces the central processor as the central resource, while packets are the units of work requiring this resource, just as jobs must compete for use of the processor. With this analogy, most real-time scheduling methods are immediately applicable to the scheduling of packets on a link.

A *scheduler* allocates the usage of a link according to some predefined allocation discipline. This discipline may be optimized for uniformity as in Round-Robin, simplicity as in FCFS, or several other criteria as in Priority-Based. Priorities may be designated by the end-user, or may be assigned according to some properties of the packet, such as the arrival period or deadline. In addition, priorities may be statically assigned for all packets in a connection, or may be assigned dynamically at the time of arrival of a packet. The scheduler may enforce priorities at the completion of the current transmission, or may elect to preempt an active transmission in favor of a newly arrived packet. These are called non-preemptive and preemptive schedulers, respectively.

As described in section 2, hard real-time traffic is often periodic. The period of a connection i is the interval between the arrival of successive packets, and is denoted T_i; the transmission time of each packet in i is denoted τ_i, and the end-to-end deadline is D_i. The *due-date* for a packet (or simply the deadline) is the sum of its arrival time and its end-to-end deadline[10] Dynamic preemptive

[9] Nodes that operate at the link layer are also termed switches in the literature for high-speed networks.

[10] In our usage, the synonymous terms "due-date" and "deadline" are time instants, while the synonymous terms "maximum allowable latency" and "end-to-end deadline" are time intervals.

schedulers *Earliest Due Date* or EDD (also called Earliest Deadline First (EDF)) [77] are preferred in cases where individual link delays must be less than the packet interarrival time [5, 40, 63, 121]. In the EDD method, the packet with the earliest due date has the highest scheduling priority.

To guarantee that user-specified end-to-end deadlines can be met, the *schedulability* of individual links must be checked. A set of real-time connections is schedulable on a link if it can be guaranteed that no packets in those connections will miss their deadlines on that link. When the EDD scheduling discipline is used and the link deadline for every packet is equal to the packet interarrival time for its connection, the connections are schedulable as long as link utilization is less than 100%. When EDD is used but link deadlines can be less than packet interarrival times, schedulability checking is much more difficult. The complexity of schedulability checking in this case is proportional to the product of the periods of all connections using the link.

3.3 Traffic Characterization

A hard real-time application requires a specific quality of service from the network; this QOS consists of delay, jitter, and loss bounds. The characteristics of the traffic generated by the application must be known in advance in order to guarantee this quality of service. A prediction of the exact arrival time and length of every packet could be used for this purpose. However, this requires perfect knowledge of future behavior, which is not possible for variable-bit-rate sources. Instead, several different models of traffic have been proposed. These models are statistical in nature and so do not require precise knowledge of the future. They are also amenable to calculation of the resources required to provide a guaranteed quality of service.

The traffic characterization used by most hard real-time communication methods is the peak rate model. The parameters of this model for each connection i are the minimum inter-arrival time T_i, the maximum packet length τ_i, and the delay bound or end-to-end deadline D_i. The bandwidth or rate requirements for such a connection are τ_i/T_i bits per second; we use the variable ρ_i to symbolize this rate. The peak rate model is exact only for constant bit-rate traffic; it overstates bandwidth needs for all variable bit-rate sources.

The Linear Bounded Arrival Process model (LBAP) [28] uses as an additional parameter the maximum burst size σ_i. In this model, in any time interval t the maximum number of arriving packets may not exceed $\sigma_i + (t/T_i)$. Deterministic delay bounds can be specified and met for this model. The leaky bucket [118] implements LBAP by defining a bucket containing up to σ_i tokens. Additional tokens are generated every T_i seconds. For each arriving packet, one token is taken out of the bucket. When an arriving packet finds an empty bucket, it can be discarded or queued; in either case, it is not allowed to enter the network immediately upon its arrival.

Golestani [43] characterizes a connection by its rate r_i and its frame F, with interval T_F. A traffic source is permitted to generate no more than $r_i \cdot T_F$ bits during any interval of length T_F. There are only a limited set of frame intervals available for the user to choose from. Lea [72] also advocated limiting the set of allowable rates, as it simplifies the tasks of capacity planning and routing. Simulation results indicated that the capacity losses due to oversubscription, i.e., specifying the higher rate for a traffic source whose rate is midway between two quantized rates,

Table 1: NOTATION FOR PACKET SCHEDULING PARAMETERS

Symbol	Interpretation
e_i^j	Time ith packet is eligible to be sent on ℓ^j
d_i^j	Delay bound for ith packet on ℓ^j
a_i^j	Time last bit of the ith packet is transmitted on ℓ^j
κ_i^j	Slack $\kappa_i^j = d_i^j - \tau_i$ on ℓ^j

were not very great.

These models have generally been developed on the basis of simplicity and tractability for analysis. There is a lively debate concerning the accuracy of these models, and the application's ability to determine *a priori* the correct parameter values.

3.4 Connection-Level Processing

The network is responsible for establishing a real-time connection between two user applications, and for ensuring that the connection is reliable and provides satisfactory service. The functions that must be performed include routing, admission control, error correction, and flow control. There is little published work specific to routing of real-time traffic.[11] Error control for real-time communication is another relatively unexplored but important problem. For most real-time applications flow control is not required, as the destination decoder is designed to keep up with the source encoder data rate. In this section we describe the remaining function of connection-level processing, which is admission control. We do not include in this discussion the reservation of resources such as processing bandwidth at the destination, as that is outside of the network.

The purpose of admission control is to calculate which network resources are required to provide the QOS requested by a connection, determine if those resources are available, and then reserve those resources. The resources that need to be reserved are primarily buffer space at each node and either bandwidth for each link along the connection's path.

Admission control typically proceeds in two phases. The first phase determines if the resources needed at each node along the path are available. The second phase allocates these resources to the connection if the first phase is successful. The first phase propagates in a "forward" direction from the source to the destination, while the second phase propagates "backwards" to the source. If a connection is not admitted at the requested QOS, the application can choose to renegotiate at a lower QOS. For hard real-time applications, this means extending the end-to-end deadline, relaxing the jitter requirements, or decreasing the peak traffic rate or permissible burst size. For some networks, the application may also be able to try another route. More detailed descriptions of real-time resource allocation protocols which have been implemented can be found in [4, 34].

In table 1, the notation used to describe the various methods is defined.

[11] Zheng and Shin [133] propose but do not investigate a method based on link deadlines, suitable for deadline-based scheduling algorithms.

Scheduler-based methods The scheduler-based methods for hard real-time communication are:

EDD-D: earliest due-date for delay [40]

EDD-J: earliest due-date for jitter [121]

SRT: smallest response time [63]

PCT: preemptive cut through [5]

An application specifies the end-to-end deadline, D_i, for the packets in connection i. From this deadline for path Γ_i, the link deadline d_i^j for each link $\ell^j \in \Gamma_i$ must be determined during the first phase of admission control. Ferrari and Verma [40] were the first to propose a method for computing the minimum acceptable link deadline. For each link the feasibility of scheduling the existing connections plus this new connection i must be checked. Their method is only valid under the assumption that the sum of all packet transmission times is less than the shortest period of any connection using the link. Kandlur [63] removed this restriction with an algorithm that assigns static priorities to existing connections based on their link deadlines. The static priority assignment results in non-minimal deadline assignments in certain circumstances. Zheng and Shin [133] proposed an algorithm for this same purpose which is more complex, but is locally optimal. [12] If there is no feasible schedule on one of the links in Γ_i, the new connection is denied admission to the network at the requested quality of service. In addition, the sum of achievable link deadlines must be less than or equal to the end-to-end deadline.

The second phase of admission control allocates the bandwidths and deadline intervals required for the connection to meet its end-to-end deadline. This phase can also relax resource allocations when the requested end-to-end QOS has been exceeded. For the scheduler-based methods, this works as follows. Let the end-to-end slack be equal to the difference between the offered and required end-to-end deadlines. Dividing this slack among the links of the path allows the deadline requirements of future connections to be more easily satisfied. Ferrari [40] suggested evenly dividing the end-to-end slack among all of the links on the path. Aras [5] suggested an adaptive admission algorithm which allocates slack to the more heavily congested links. Simulation results indicated that this algorithm permits higher utilization with tighter end-to-end deadlines than Ferrari's approach.

Rate-based methods The rate-based methods of hard real-time communication are:

HRR: hierarchical round robin [62]

S&G: stop-and-go [43]

WFQ weighted fair queueing [33] (and the similar **PGPS**, or packet generalized processor sharing [94])

[12] There is no known technique for determining the d_i^j's for any measure of global optimality, such as network utilization.

RCSP: rate-controlled static priority [129]

We now describe the first phase of admission control for the rate-based methods, starting with RCSP. In that method, a new connection is assigned a target link deadline on each link along its path. For each link, the connection is assigned a scheduling priority according to its link deadline, where small link deadlines $\Leftrightarrow$ low priority number $\Leftrightarrow$ high priority. A new connection with priority number h can only affect the delay bounds of connections with lower priority. A simple computation for each priority number greater than or equal to h is sufficient to ascertain if this new connection can meet its delay bound without causing other connections to miss their deadlines.

S&G and HRR provide the easiest means of admission control. Let T_{F_i} represent the period of the frame size for connection i. A simple bandwidth check $(\sum_{k \in \Omega^j} \tau_k / T_{F_k} \leq 1)$ is all that is necessary to determine if connection i can be successfully scheduled on link ℓ^j. If connection i can be scheduled on all the links along its path, then it can be admitted to the network.

3.5 Per-Packet Processing

Each packet of an admitted connection is conveyed through the network along the path established for that connection. At a switching node, the packet is multiplexed onto the next link along its path, along with packets of other connections using the same link. In this section we describe the various methods of multiplexing hard real-time traffic onto a link. In this discussion we do not address the separate problem of switch contention, which affects both real-time and non-real-time traffic equally.

Information about an admitted connection is stored at each node along the path of that connection. This information we will call a *descriptor*. The descriptor must contain data such as packet periods/interarrival times, maximum lengths, service quanta or rates, maximum burst size, link deadlines, and resources allocated to the connection. Each incoming packet must contain a connection ID as part of its header.

To unify our discussion we present a simple model of the real-time processing performed at each output link of a node. This model is depicted in Figure 3. The steps of processing are:

- Input regulation, which shapes the input arrival characteristics
- Packet demultiplexing, which inserts a packet into one of a set of queues, corresponding to different QOS guarantees
- Queue insertion, which is either FCFS or priority-based
- Queue multiplexing, which selects the next queue to service, and how many packets to remove and transmit from that queue

A scheduling policy can be classified as either *work-conserving* or *non-work-conserving*. A method is work-conserving if an output link will never be idle as long as there are packets waiting to use that link. Work conservation might seem attractive, since it promises lower average end-to-end delays for packets. However, methods which minimize jitter are always non-work-conserving.

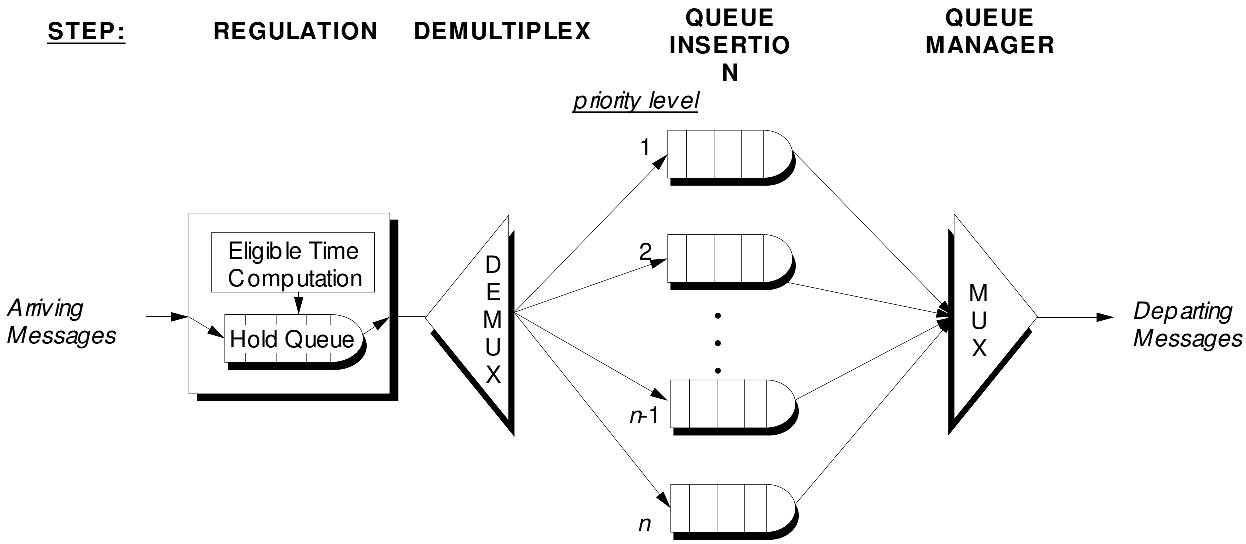

Figure 3: Node processing per output link

For hard real-time traffic, reducing the jitter and maximum packet delay is usually more important than reducing the average packet delay.

3.5.1 Scheduler-based Methods

For all of the methods, guarantees on delay and jitter can only be provided if the input arrival process conforms to a particular model; this was described in section 2. Unfortunately, even packets arriving at the edge of the network obey this model perfectly, the arrival process at an intermediate node along the path may not conform to the model. A regulator smoothes the packet arrival process at the intermediate nodes by delaying the arrival of early packets. The logical arrival time $l_i^j(k)$ for the kth packet of connection i is defined to be the greater of its actual arrival time a_i^j, and the sum of the previous logical arrival time plus the period of the connection. By this definition, the difference between succeeding logical arrival times will always be greater than or equal to the period of the connection. A packet can be held by the regulator until its logical arrival time; this restores the expected traffic behavior for that connection.

Without further regulation, delay jitter can still be quite high. EDD-J was developed to address this problem. The regulator is required to totally eliminate the internal jitter due to queueing delay; that is, the regulator restores the traffic source's original arrival pattern at every node along the path of the connection. Let the holding time for a packet on link ℓ^j be defined as the time the packet spends in the input regulator's queue. This holding time is calculated as the difference between the departure time of that packet from the previous node, and its link deadline at that previous node. The time at which the packet becomes eligible for transmission is the sum of its arrival time plus this holding time. The regulator thus absorbs delay variations by holding a packet for the amount of time it left the previous node ahead of schedule. The due date is the maximum of the eligibility time plus the link deadline for this node, and the logical arrival time for this node.[13]

Since all of the scheduler-based methods use a single output queue for hard real-time traffic, there is no need for queue demultiplexing or multiplexing.

Packet insertion into the output queue is based on priority for these methods; this priority is determined by earliest due-date. The due-date for a packet is equal to the sum of its logical arrival time and its link deadline for this node. As long as input traffic conforms to the model parameters negotiated by each connection, every packet will meet its end-to-end deadline.

The scheduler-based methods all use a packet as the unit of scheduling. A packet being transmitted can be preempted by a newly arrived packet with higher priority, i.e., earlier due-date; a preempted packet will have its transmission resumed when all packets with higher priority have finished their transmission. Using long packets reduces scheduling overhead, while using shorter packets reduces or eliminates the possibility of preemptions. In addition, end-to-end latencies are increased by using longer packets. The preemptive cut-through (PCT) data transfer protocol [5] is a variation of EDD-J that offers much lower end-to-end delay. In PCT, the transmission of a packet is pipelined over multiple links along its path. PCT can achieve an end-to-end delay close to that of circuit-switching if link deadlines are set to their minimum possible values Kandlur [63] has proposed another method of splitting long packets in order to pipeline their transmission.

3.5.2 Rate-Based Methods

The rate-based methods exhibit a greater variety of mechanisms than is the case for the scheduler-based methods; therefore, we discuss packet multiplexing for each method individually.

The Stop-and-Go method (S&G) schedules packets as groups by clustering them into frames. For each connection i assigned to a frame F, the frame size T_F is stored in its connection descriptor. There is also conceptually a clock for each frame size, which emits a signal every T_F seconds. When a packet arrives, it is buffered until the clock for its frame emits its signal. The maximum holding time is bounded by the phase mismatch of the frame clocks at successive nodes. When the signal occurs, all packets buffered for the frame are transferred to an output queue. Each output queue implements a FCFS policy. The output queues are multiplexed in priority order, with shorter frame sizes having higher priority. All eligible packets in higher priority queues are transmitted before a packet in a lower priority queue will be transmitted. Since the residence time of a packet at a node is constant, the jitter is limited to only the last link, and is no greater than $2T_F$. With no phase mismatches, end-to-end delay with this method is H_iT_F; with the worst-case phase mismatch at every node along the path, end-to-end delay as high as $2H_iT_F$.

HRR is conceptually similar to S&G in that packets are grouped into frames for scheduling purposes. Each connection is assigned to one of g fixed rate levels, where level 1 is the highest rate level. Each level k corresponds to a frame of size n_k slots. The frame for the kth level starts transmission every F_{Tk} seconds, where $F_{T1} < F_{T2} < \ldots < F_{Tg}$. A connection i which is assigned

[13]This explanation is for the case in which no jitter is allowed in the network. The calculations can be easily modified in the case where some jitter is allowable at each node along the connection's path.

to a level k is allocated s_i slots out of each n_k slots alloted to that frame. As packets for connection i arrive, the input regulator will only release s_i of them for transmission during each interval of F_{Tk} seconds. Thus the rate allocated to connection i assigned to level k is s_i/F_{Tk} slots/sec. There is one FCFS output queue for each level. For each frame of size n_k, b_k slots are reserved for frames with lower rates/priorities. Packets at lower priority levels than k are transmitted after the first $n_k - b_k$ slots of the frame for level k; HRR is thus non-work-conserving. End-to-end delay and jitter are both less than or equal to $2HF_{Tk}$ for a connection assigned to level k. Banerjea [8] analyzed the queueing delays in some detail.

In weighted head-of-the-line processor sharing, each connection has a separate queue, and the first packet of each queue gets a weighted fair share of the bandwidth. Parekh and Gallager [94] showed that for networks where the sources are leaky bucket constrained and where nodes approximate the weighted head-of-the-line processor sharing service discipline on a packet-per-packet basis, the end-to-end delay can be bounded tightly. For networks such as ATM with fixed-length packets, this processor sharing discipline is equivalent to weighted round-robin scheduling. For networks with variable-length packets, WFQ [33, 44] can be used. WFQ simulates processor sharing by scheduling packets for transmission in the order of their finishing time under true bit-by-bit processor sharing. For each arriving packet, the scheduler needs to determine the finish time under processor sharing and insert the arrival into a priority queue. The bound on the queueing delay in a H-hop network can be expressed succinctly if all connections are allocated a share of bandwidth proportional to their ρ_i's:

$$D \leq \frac{\sigma_i + H_i \tau_i}{\rho_i} \tag{1}$$

For hard real-time traffic, the bound on the queueing delay in an H_i hop path falls between the lower and upper delay bounds possible for stop-and-go queueing. This relationship again emphasizes the connection between policies that yield deterministic delay bounds.

The RCSP method [129] can use either of two regulators; we discuss only the delay jitter control regulator here. The eligible time e_i^j of an incoming packet is defined as $e_i^j = e_i^{j-1} + d_i^{j-1}$. An early arriving packet is simply held until the time at which a latest-possible packet would have arrived; this is similar to the EDD-J regulator. There are a set of FCFS output queues, one for each possible priority level. The priority level is determined statically based on earliest deadline. The queues are multiplexed by always selecting for service the highest-priority non-empty queue. RCSP can guarantee delay and tight jitter bounds. However, it is not clear how to choose the link deadlines to achieve a specific quality of service.

3.5.3 Implementation Requirements

In this section we assess the implementation complexity of packet multiplexing. The implementation requirements of admission control are not addressed. Since admission control is relatively infrequent and needs to be flexible, it is better implemented in software than in hardware.

A very important issue for communication networks is the required amount of buffer space. The buffer space is generally closely related to the maximum stay of a packet at a node. Table 2

Table 2: BUFFER SPACE REQUIREMENTS

Method	**Maximum Residence Time**
EDD-Delay	$\sum_{l=1}^{h} d_i^l$
SRT	$\sum_{l=1}^{h} d_i^l$
EDD-Jitter	$d_i^{h-1} + d_i^h$
PCT	$\kappa_i^{h-1} + d_i^h$
RCSP (delay-jitter)	$d_i^{h-1} + d_i^h$
RCSP (rate-jitter)	$\sum_{l=1}^{h} d_i^l$
S&G	$2T_i + \theta_i$
HRR	$2T_i$

summarizes the buffering requirements of the methods, for each node along a connection's path. In the table, h denotes the number of links traversed or hop-count from the source of the connection to the node. The non-work-conserving methods require less buffer space than work-conserving ones because non-work-conserving disciplines reduce the delay-jitter at each link; thus, the residence time of a packet on each link is fairly uniform. In contrast, for work-conserving methods the maximum possible residence time of a packet increases linearly along its path. Thus larger buffers are needed at nodes farther and farther "downstream."

The regulator holds incoming packets in a buffer until their proper release time. This buffer can be implemented by a calendar queue [17] and a single reference clock. For queues serving packets with small delay bound variance, the average insertion time of the calendar queue is $O(1)$. For the HRR and S&G methods, an alternate implementation is to have one FCFS queue per frame size, with one reference clock for each queue.

The output queue for scheduler-based methods is implemented as a priority queue. Priority queues are somewhat expensive, both in hardware cost and processing time; see [21] for a discussion of different ways to implement priority queues. For the rate-based methods output queues can be implemented as FIFOs, which are simpler and faster.

A queue demultiplexor can be implemented with simple combinational logic. For S&G and RCSP, the non-empty queues (one per frame or rate) are multiplexed based on their priority; a priority encoder is needed for this purpose. For HRR, frame start times are fixed, so queue multiplexing is simpler.

Another source of overhead which should be acknowledged is the number of bits in the packet header which are required by a method. The jitter control algorithms (EDD-J, SRT, RCSP, and PCT) require a timestamp in each packet, which is updated at each node along the path.

Table 3: HARD REAL-TIME COMMUNICATION SERVICE DISCIPLINES

Method	Type	$D_{\min}$	$D_{\max}$	J
EDD-Delay [4,40,63]	Scheduler	$H\tau$	HT	$D - H\tau$
EDD-Jitter [121]	Scheduler	$H\tau$	HT	$T - \tau$
PCT [5]	Scheduler	τ	HT	$T - \tau$
RCSP-Rate [129]	Rate	$H\tau$	HT	$D - H\tau$
Stop-and-Go [43]	Rate	HT	$2HT$	$2T$
HRR [62]	Rate	$H\tau$	$2HT$	$2HT - H\tau$
WFQ [25],PGPS [94]	Rate		$HT + H\tau_{\max}$	$D - H\tau$

3.6 Summary

We have presented a variety of methods for hard real-time communication. All of these methods offer a quality of service which has not heretofore been available from packet-switched networks. None of them is clearly superior in all respects. Table 3 summarizes the delay and jitter characteristics of the hard real-time communication methods. In this table, H stands for the number of hops on a connection's path, and T stands for the packet interarrival period. $D_{\min}$ and $D_{\max}$ denote the minimum and maximum value of the delay bound that could be requested from the method.

Jitter and buffer space are minimized by the EDD-J method of Verma [121] and its derivatives (SRT, RCSP, PCT). End-to-end latency is minimized by PCT. Implementation is more straightforward for the rate-based methods, although scheduler-based methods also appear to be practical. An important point to make about rate-based methods is the coupling between the service parameters. For instance, Golestani shows that for the Stop-and-Go method [43], jitter, buffer space, and end-to-end delay are all linearly proportional to the frame size T_F, and the increments of bandwidth allocation are inversely proportional to T_F. A similar coupling exists in round-robin methods such as HRR. In contrast, delay and bandwidth requirements are satisfied independently by scheduler-based methods.

We have discussed only point-to-point networks in this section. However, there are many real-time applications which may need to run only on a single LAN. In addition, wide-area connections will frequently span one or more local area networks, in addition to the long-distance links and network switches. Clearly, it is important that LANs also provide real-time communication services.

Recent research on LAN real-time services has concentrated on the Token Ring and FDDI. A real-time service called the timed token protocol [2] can be implemented in these networks. Unfortunately, this protocol can only support a restrictive set of delay bounds. Strosnider [115] and Lim et al. [74] applied the earliest deadline first scheduling technique to extend real-time support for arbitrary delay bounds. However, the maximum network utilization in FDDI is limited to 33 percent, and protocol overheads are very high. Buffer insertion rings like ORBIT [24] allow scheduling of packets on each station. Zheng [132] proposed a method based on EDF scheduling; this method results in lower overheads and allows full network utilization. For a detailed survey of other multi-access real-time communication protocols, we refer readers to a survey by Malcolm [80].

4 Soft Real-Time Communication

As discussed in section 1, soft real-time applications such as interactive packetized voice and video can sustain a certain amount of packet loss without significantly affecting the overall communication "quality". Packet loss can result either from buffer overflow at the destination or within the network, or from late packet arrivals at the destination. For short audio segments, tolerable loss values as high as 50 percent have been cited [82], while high-quality audio has been shown in subjective tests to tolerate loss rates of five percent for speech and ten percent for music [84]. Tolerable losses for video are generally much lower, but depending on the coding algorithms used and the effort expended on reconstructing lost video cells at the receiver, packet losses of as much as one percent can be sustained [65]. Loss tolerance is higher if the source can designate particular packets for preferred dropping; this is termed hierarchical coding.

In this section, we discuss network architecture and protocol mechanisms designed specifically for handling such loss-tolerant *soft* real-time traffic. As we will see, the ability of these applications to tolerate a certain amount of traffic loss allows a richer set of network- and application-level control mechanisms to be considered. We begin our discussion at the application layer and then work our way "down" the network architecture.

4.1 Application-level Characteristics

As in the case of the hard real-time applications, soft real-time applications will likely need a certain guaranteed quality of service before being admitted to the network. In the case of soft real-time communication, the QOS requirement will be the delay and jitter bounds, and the application's maximum tolerable packet loss due to either buffer overflow or exceeding the delay bound [70]. To determine the QOS which can be offered to an application, the network must first characterize the application's traffic characteristics. The two dominant classes of soft real-time traffic which we discuss are audio and video.

The traffic characteristics of soft real-time applications can vary over time. An example is packetized voice. In the case of voice sources, the variation results primarily from the "on-off" characteristics of human speech. While a speaker is talking, packets are periodically generated. During periods of no speech, such as pauses between words and sentences, the packet generation rate may change [35]. The decision of whether or not to generate packets during periods of silence, and indeed the definition of silence periods themselves, is application-dependent.

The statistics of these silence and talkspurt periods have been studied for conversational voice [48, 60, 82, 91]. A number of Markovian models for voice packet generation have been proposed for interactive conversations, both for a system with two parties [14,15] and for a single party [13,82]. The most common model for a single party is that of a two-state (silence and talkspurt) Markov chain [15,60,82]. In this model, a speaker talks, generating packets periodically, for an exponentially distributed amount of time, and then becomes silent for another exponentially distributed amount of time. Successive talkspurt and silence periods lengths are assumed to be statistically independent. It has been recognized [126], however, that these models may not accurately capture voice patterns and in particular, the silence periods . A three-state model with long and short silence periods has been suggested [16]. Models for voice monologues, such as lectures, [9, 50, 114, 122] depend strongly on the sensitivity of the silence detector. Recent research [108] indicates that not only are silent periods poorly modeled by exponential distributions, but also the delays predicted by that model significantly underestimate actual network delays, even when several sources are multiplexed.

The leaky bucket model described in section 3.3 is a popular model which requires characterization of the bucket capacity and token generation rate. Without cooperation from the source, it may be difficult to find a descriptor where the rate ρ is less than the peak rate. For voice with silence suppression, for example, the token bucket capacity is determined by the maximum talk spurt duration, which is usually not known in advance.

Characterization of video sources is even more problematic than voice, for one because the traffic characteristics depend strongly on the video coding algorithm employed. Statistical source descriptions have been attempted for quite some time [22, 51, 53, 75, 89, 93, 109]. Most descriptions focus on the luminance portion, since it dominates the bandwidth requirements and is considered to be representative of the whole video signal. For example, it was found that either a gamma [53,109] or a normal distribution [79] describes the bit generation process. A normal distribution may also be an appropriate model for the aggregate bit rate of ten or more sources [127].

Two potential uses for model-based video source descriptions are for the generation of simulation sequences and for use in analytical performance models. For the former, autoregressive models of differing complexity have been widely used. First order models [79, 89, 120, 128] and second-order models [53] can capture the autocorrelation structure of video sources. The exponential decay of the autocorrelation function implied by autoregressive models may depend on the source material [55]; for example, the rhythmic head movements of singers are clearly reflected in a periodic autocorrelation function. Beyond the source correlation, the queueing delay is also affected by rapid changes in bit rate during scene changes. Models combining several autoregressive processes try to capture this effect [100, 101, 127]. A somewhat more general random process known as TES has also been used [73, 81]. The influence of the codec and source material on a range of statistical measures, in particular entropy-related ones, is shown in [103].

As we will see in the following section, connection admission decisions are based on the assumptions that the traffic characteristics of the existing and arriving connections are sufficiently well known and that the sources indeed conform to these characteristics. Generally, sources have to be more closely controlled for tighter quality of service commitments and more highly-utilized networks. Traffic can be shaped by passing it through a device that delays packets to ensure, for example, that the advertised peak rate is not exceeded [12, 47, 129]. The source itself may be able to adjust its transmission rate to changing network or receiver conditions, although the rate is fixed for many current audio and video codecs. The network may also need to protect itself from malfunctioning or malicious traffic sources. It does this by dropping or marking packets or connections when the agreed-upon traffic characteristics are violated. The latter action is commonly known as policing. Both shaping and policing may be used within the same network.

4.2 Connection-level Issues

As in the case of hard real-time applications, the most important connection-level issue is whether or not a soft real-time connection can be admitted to the network at its requested quality of service. It should be noted that soft real-time applications do not, by their nature, require that a QOS guarantee be provided. Indeed a number of recent experiments [18] have demonstrated the possibility of supporting soft real-time applications over networks such as the Internet which provide no QOS guarantees. However, it appears that the ITU-TS is moving towards a network architecture that can provide strict quality of service guarantees for voice and video in ATM networks [49, p. 31].[14]

In section 4.2.2 we discuss recent research addressing the connection-acceptance and QOS issues. First, however, we consider the fact that an integrated network architecture must support not only soft real-time applications, but potentially hard real-time applications and best-effort applications as well. In section 4.2.1 , we thus discuss the larger framework in which connection acceptance decisions must be made, and survey efforts which explicitly consider the need for a network to provide support for a heterogeneous mix of applications.

4.2.1 Multiple Traffic Classes and Grades of Service

A number of proposals have been put forth to provide network support for diverse application requirements. Generally, a priority mechanism gives priority to traffic with deterministic delay bounds, followed by traffic with statistical bounds, and finally best-effort traffic. Priorities also simplify the decision of whether to accept a new connection. This is because the admission procedure for higher-priority traffic can ignore lower-priority traffic, provided enough aggregate bandwidth is left so that the QOS guaranteed to lower-priority traffic can be met.

ATS [58] offers guarantees to classes of traffic, rather than individual connections. That is, all connections within a class get the same QOS. Class I traffic experiences bounded delay and is given priority over all other classes by being able to claim all available bandwidth within a scheduling cycle. Class II may suffer some late loss, and class III is best effort. The guarantees are based on precomputing, through simulation, so-called schedulable regions that delineate the combinations of the number of class I, II and III connections that can be supported within the desired guarantees. It is important to note that the shape of the schedulable region depends on the source traffic characteristics. Either the traffic offered to the network must be predictable, or its worst-case behavior must be defined and enforced.

Tenet [38,39] aims to provide connection-specific QOS, divided into three classes: deterministic guarantees with delay and delay jitter bounds, statistical (delay bounds with acceptable delay-loss probability), and best-effort. At each node, traffic is processed by multi-class earliest due-date scheduling, with class priority decreasing from deterministic to best-effort traffic. Admission control is based on peak rates for connections with deterministic guarantees, and peak and average rates for connections with statistical guarantees. Connections are set up through the real-time channel

[14] Quality of service guarantees are particularly important to paying customers in public networks.

administration protocol (RCAP) [78], while data is transported in IP-like packets with an added channel identifier and jitter correction factor.

Sriram [111] combines the notions of traffic classes and per-connection guarantees in a round-robin scheduler. High-bandwidth CBR connections with stringent performance requirements use their own queue with individual time slice assignments, while other connections may be combined into a single assignment. Connections are admitted if a model based on on/off sources approximated by the first two moments predicts sufficient QOS.

Clark et al. [25] propose a three-level hierarchy: guaranteed, predicted, and best-effort service. Guaranteed service with deterministic delay bounds is provided by weighted fair queueing. A connection requests a particular clock rate based on worst-case queueing delay it can accept. The connection will be accepted if there is sufficient remaining capacity at every link along its path to accommodate its assigned clock rate. Predicted service uses the bandwidth not allocated to guaranteed service; admission control for predicted service is not precisely defined. Predicted service uses FIFO+ scheduling with several priority classes to reduce delay variance for multi-hop connections. FIFO+ increases the scheduling priority of packets that have experienced delays above the average for their class. Best-effort traffic is assigned the lowest priority, isolating all other classes of traffic from it. Clark advocates reserving a fixed minimum bandwidth for this class.

Resource reservation for real-time communication and the actual data transfer can be combined into a single protocol or split into two protocols. The Internet ST-II protocol [96,117] is an example of a combined protocol. It tries to accommodate a variety of resource management policies by simply conveying a flow descriptor from a source to the destination(s); resources are reserved and a virtual circuit is set up along the way. The protocol itself does not specify or support packet scheduling. The SRP resource reservation protocol [3], on the other hand, is an example of a split protocol. It uses a remote-procedure-call mechanism to reserve resources, but does not carry user data.

As long as packets within the network can be reordered or experience variable delays, isochronous applications require an end-to-end mechanism to reconstruct the source timing relationships between packets. Protocols for voice transport [27] and more general real-time transport protocols [32, 104, 123] address this need.

4.2.2 Providing Statistical Guarantees on Delay and Loss

A number of researchers have advocated providing soft real-time applications with connection-level statistical guarantees on packet loss. In this section, we briefly describe three different approaches to provide such statistical guarantees. These are the source-based, bounding, and observation-based approaches. Additional information about some of these techniques may be found in [70].

Source-based approach In the source-based approach to providing QOS guarantees [37,40,46, 124], traffic sources at the network's edge and within the network are characterized by relatively "simple" models. An example of such a source model is the on/off voice source [46,85,124] described in section 4.1. In order to determine whether or not the multiplexed sources will receive their required QOS, the queueing behavior of the multiplexed traffic sources is then analyzed. In [37,46], the QOS measure of interest is packet loss; in [40] the measure of interest is maximum delay.

One advantage of the source-based approach is its simplicity, which makes it well-suited for real-time, on-line implementation. For example, the connection admission control mechanism based on the approximate QOS scheme described in [46] can make a QOS computation with a very small number of additions and multiplications. Source-based soft real-time guarantees can be fulfilled using simple disciplines such as FCFS. When hard real-time guarantees are required, a more complex scheduling discipline is required, as described in section 3.2. Finally, unlike the case of hard real-time traffic, soft real-time guarantees can be made when the aggregate peak rates exceed the link capacity.

There are several open issues regarding source-based models, however. The first issue is the extent to which more complicated sources can be characterized by the relatively simple source models considered thus far. A more fundamental concern is that the traffic models employed, whether at the source or deep within the network, require some form of Markovian assumptions. While traffic at the edge of the network may be reasonably well-approximated by such models [85], it is still unknown whether this is also true for a connection's traffic when it is "deep" within the network, where the traffic characteristics have potentially been altered as a result of the traffic having passed through several multiplexers. The extent to which these interactions must be considered and the extent to which a reliable guarantee can be provided without taking such considerations into account remains an important question for future research. It has been observed [11, 92] that an estimate of the worst-case performance can be obtained by assuming that a connection's traffic maintains its input characteristics as it progresses through the network.

A final open issue that arises with both the source-based and the other two approaches is that the guarantees provided are local, i.e., performance guarantees are provided to a connection at a single multiplexing point. User-specified QOS requirements, however, are based on an end-to-end performance requirement. The manner in which these end-to-end requirements are to be divided into local performance requirements which together satisfy the end-to-end requirement remains another important open research issue. An example might be to have more congested nodes provide a poorer QOS guarantee, while less congested nodes provide a more stringent performance guarantee. Some early research addressing this issue are [40, 86].

Bounding approach The bounding approach explicitly considers the effects of multiplexing on a connection's traffic characteristics, and hence its performance, as the traffic passes through various multiplexers. We illustrate the statistical bounding approach by briefly considering the methodology described in [71]. In that work, no assumptions are made about the actual cell interarrival times, as is done in traditional queueing theory. Rather, for each connection, a stochastic bound on the number of arrivals in *any* interval of time of length k is specified, typically for a set of values for k. Given these stochastic bounds on traffic at the edge of the network, bounds can then be computed for each connection's traffic after it passes through each multiplexer along its path in the network. Given a characterization of all sources at the "edge" of a given network and given the routing of connection, the process of computing performance bounds on a connection-level basis is a two-step process. In the first step, all connection flows are characterized at each multiplexer; in the second step performance bounds are computed. The two-step procedure is similar in spirit to [28, 29], although quite different in what is actually computed during each step. In [71], performance bounds on the per-connection distribution of delay are computed for a sample 27-connection 13-node network, and are shown to be tight for some traffic parameter values but quite loose for others. An important outstanding research issue for the statistical bounding approach is the extent to which traffic can be characterized by, or policed to conform to, the form of the distributional bounds required by [19, 20, 71, 125].

Perhaps the most important outstanding research issue for the statistical bounding approach is its reliance on the ability to bound the maximum length of each queue's busy period for a given set of traffic specifications. If this condition is not satisfied, no bound can be computed, even though it may be known via traditional queueing analysis that the queues themselves are indeed all stable (i.e., the expected delay at all queues is finite).

Observation based approach The final approach to providing QOS guarantees is the "observation-based" approach [25, 58, 59, 61]. In [58, 59], previously-made measurements of certain types of traffic sources are used to characterize an arriving connection and in determining the connection acceptance decision. This has the advantage of not requiring that the connection specify its traffic parameters. However, the connection must belong to one of a predefined set of classes, and its traffic must, presumably, correspond to the traffic characteristics of that class if the guarantees are to be reliable.

In the on-line approach described in [25, 61], the bandwidth requirements of already-admitted token bucket-controlled connections are determined from the current, measured behavior of these connections rather than the traffic parameters declared by these connections when they first arrived to the network. This measured behavior, together with the declared parameters of an arriving connection, are then used in making the connection acceptance/rejection decision for the incoming connection. Note that with the observation-based approach of [25, 61] no firm QOS guarantees can be made; this is because the QOS "guarantee" is based on traffic loads measured at connection admission time, and these loads may change once the connection is admitted. For this reason, connections receiving guarantees based on observation are referred to as receiving "predicted service."

A potential advantage of offering predicted rather than guaranteed service is that the network may be more fully utilized. A quantitative discussion of this issue can be found in [70]. In [61], a simulation study of a two-hop network with predictive service also indicates that the approach may indeed provide relatively reliable guarantees. A number of open research issues remain to be addressed, however, including the effects of different measurement/estimation techniques on the protocol, the overhead involved in measurement, the influence of the number of multiplexed connections on the reliability of the guarantees, and a thorough study of the mechanism in a larger network environment.

4.2.3 Best-Effort Delivery

Instead of requiring that a network provide explicit support for soft real-time applications, an alternative is to simply use an existing packet network. In this approach, all packets are typically scheduled first-in, first-out and real-time traffic is treated no differently than other traffic. For existing networks whose internal structure is difficult or impossible to modify, this approach may be the only feasible one. Usable performance can be obtained if the network is sufficiently overdimensioned and the end-user applications can adapt to variable network delays. Parts of the current Internet are examples of such a system. A limited amount of real-time voice and video has been transmitted over the Internet with some success to several continents [18, 105].

4.2.4 Synchronization of Soft Real-Time Traffic

The audio and video applications previously discussed form an important sub-class of real-time applications. In these applications, the receiver is expected to deliver data a fixed amount of time after its generation to the destination application, reconstructing the timing pattern at the sender exactly. In telephony parlance, these applications are referred to as *isochronous.* Unless packets traversing the network experience deterministic delays, isochronous applications have to delay packets that arrive before their deadline (also called the *playout time*) to compensate for the network delay jitter. The synchronization method depends on whether the data stream can be broken up into smaller units that can be shifted slightly with respect to each other. For strictly synchronous connections that are continuously active, a simple elastic store or queue is sufficient. If the clock used to sample the source is not strictly synchronized with that used to consume the data at the receiver, measures such as adjustable clocks [1] and digital phase-locked-loops [31], speech time scaling [41, 88] or frame dropping/replication [26] must be used.

When multiple traffic sources are present in an application, as would be the case in a multimedia application, the playout of these connections must be synchronized. The reader is referred to [76] for a recent discussion of research in this area.

4.3 Per-Packet Processing

In this section, we discuss link-level mechanisms for multiplexing packets. These mechanisms aim to improve the performance of soft real-time applications by either lowering the delay variance or reducing the probability of extremely long delays. However, they do not offer quantifiable guarantees. The essence of many of these mechanisms is to utilize individual packet deadline information to schedule packet transmissions over the outgoing link in such as way as to minimize the number of packets lost to excessive delay.

4.3.1 Priority Policies

Priority policies can affect the order of service (time priority) and determine which packets get discarded when the buffer at a queue fills up (space priority [52, 67]). For example, in current equipment for sharing leased lines, real-time voice packets receive service priority [87]. Since for some real-time services, low delays are more important than low buffer overflow losses, it may be appropriate to give space priority to data traffic and time priority to real-time traffic. Awater and Schoute [6] investigate the optimal combination of low-delay or low-loss policies through dynamic programming. When the buffer fills, the oldest low-delay packet is replaced. For a slotted system with Bernoulli arrivals, they find that a threshold policy performs best, with service priority given to the low-delay packets as long as the number of low-loss packets is below a given threshold, where the threshold depends on the desired tradeoff between loss and delay. The authors argue that for their system low average delay also translates into low delay variance. Kubota et al. [68] propose a pure space priority scheme that discards loss-tolerant audio cells first, then data traffic, and finally, loss-sensitive video traffic. Several different scheduling algorithms for providing varying degrees of priority to real-time traffic were examined in [23]. The issue of scheduling two classes of soft real-time traffic with correlated deadlines was considered in [99].

A local queue control policy is proposed in [106, 107] that discards on arrival to a queue those packets that are going to wait longer than a set time. This policy is based on the observation that during temporary overload resources are wasted in carrying packets that would likely miss their end-to-end deadline. For traffic that can tolerate losses of a few percent, the combined loss from selective discarding and excessive end-to-end delay can be cut in half for a five-node network. It may be preferable to either distinguish packets of different importance [30, 97, 98] or truncate packets under congestion, removing less-significant information first [41, 45, 54, 64, 112, 113].

An overview of other priority policies suitable for reducing delay losses is given in [7, p. 181].

4.3.2 Laxity-based policies

The asynchronous time sharing system (ATS) [59] also offers a traffic class suitable for soft real-time sources. In the MARS scheduling algorithms, the so-called class-II traffic gets the slots in a round-robin cycle not needed by the class I (delay bounded) traffic. Within the class-II traffic, the scheduler again delays packets as long as possible without violating their delay constraint. Even though the paper assigns a delay violation probability and a maximum gap length, these values are derived from simulation assuming a given traffic pattern for class I and class II traffic. It may be possible to define worst-case enforceable traffic characteristics so that, together with an appropriate connection admission policy, class II traffic would indeed receive a statistical guarantee.

Deadline-based policies such as these divide the end-to-end deadline into per-node deadlines. In many real networks, however, only a subset of nodes are congested and have trouble meeting packet deadlines. Thus, to increase node utilization and to reduce the delay variance, it has been suggested [106, 108] to use the laxity divided by the number of hops left to travel as the scheduling criterion. This metric also has the advantage that it readily gives the same delay performance to connections with large and small numbers of hops. The method has the disadvantage that the laxity measure for all packets in the queue (or at least the first few) has to be recomputed at every scheduling instant. In addition, nodes must have clocks which are carefully synchronized, or which can measure propagation delays very accurately. Hop laxity scheduling has been successfully implemented in the DARTnet test network operating at T1 rates. A simpler policy that uses only end-to-end laxity did not perform as well.

In the current hop-laxity scheduler, real-time traffic takes precedence over non-real-time traffic. It may be advantageous to apply the philosophy of the MARS scheduler and MLT to schedule best-effort traffic if all real-time traffic has per-node deadlines sufficiently far into the future.

Another approach that avoids the difficulty of using explicit deadlines was proposed by Clark et al. [25]. Their FIFO+ scheduling policy tracks the average queueing delay experienced by all packets at a particular node through a low-pass filter. On departure, the amount of time that a packet's individual delay differed from the average delay is added to a delay variance accumulator in the packet header. Packets are served in the order of this difference timestamp. For a simple network, the FIFO+ policy was shown to reduce delay variance and the 99.9 percentile delay value.

5 Conclusion

In this paper we have reviewed models and methods for real-time communication in packet-switched networks. We described both architectural and protocol aspects of hard and soft real-time communication, as well as the integration of these forms of real-time communication with each other and with non-real-time traffic. Methods such as these will play an important role in the approaching era of high-speed integrated networks. Our intent has been to unify the body of research on this topic by offering a framework in which it can be viewed.

We have also described potential topics for future investigation. We summarize some of these open problems here:

Routing We conjecture that routing algorithms which provide QOS guarantees to real-time traffic will be more similar to today's circuit-switched (telephone) routing algorithms than today's packet-switched routing algorithms. Furthermore, some measure of schedulability or connection blocking probability, rather than utilization and average delay, will be the performance metrics used to evaluate these future routing algorithms.

Fault tolerance Detection, reconfiguration, and recovery from faults is required for real-time communication in wide-area networks. Existing methods are probably not suitable because they fail to address the need of real-time applications for continuity and extremely quick response time.

Error Control Backward error correction, as practiced in existing protocols, requires sufficient time for acknowledgment timeout and retransmission to occur. The delay of wide-area networks is a serious problem for achieving real-time deadlines in such a case. The bandwidth-delay product of high-speed networks would also require enormous buffers to support this approach. Forward correction appears to be more promising, but the information overhead and processing complexity required to reach desired error probabilities must be reasonable.

Synchronization Many multimedia applications require some form of synchronization at the destination. When multiple synchronized connections are routed over a single path, the synchronization problem is the simplest. However, in some cases the separate connections of a multimedia application could follow different paths to the destination, with distinctly different delays.

Multicasting Real-time applications such as teleconferencing are prime users of wide-area multicasting. The construction of bandwidth-efficient and scalable dynamic multicast topologies is a challenging problem.

Note: The publications listed in Table 4 are available through anonymous ftp from sites on the Internet.

Table 4: FTP SITES FOR PUBLICATIONS

tenet.berkeley.edu : pub/tenet	[8, 38–40, 62, 66, 78, 95, 121, 129, 130]
ftp.csc.ncsu.edu : pub/rtcomm	[5], this paper
gaia.cs.umass.edu : pub	[23, 57, 70, 71, 85, 86, 88, 99, 104–108]
ftp.eecs.umich.edu : outgoing	[110, 131–133]

References

[1] Stephen Ades, Roy Want, and Roger Calnan. Protocols for real time voice communication on a packet local network. In *Conference Record of the International Conference on Communications (ICC)*, pages 525A–530 (17.1), Toronto, Canada, June 1986. IEEE.

[2] Gopal Agrawal, Biao Chen, and Wei Zhao. Local synchronous capacity allocation schemes for guaranteeing message deadlines with the timed token protocol. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 2b.2.1–8, San Francisco, CA, 1993.

[3] David P. Anderson. SRP: a resource reservation protocol for guaranteed-performance communication in the Internet. Report UCB/CSD 90/562, Computer Science Division, University of California at Berkeley, Berkeley, CA, February 1990.

[4] David P. Anderson, Shin-Yuan Tzou, Robert Wahbe, Ramesh Govindan, and Martin Andrews. Support for continuous media in the DASH system. In *Proceedings 10th International Conference on Distributed Computer Systems*, pages 54–61, Paris, France, May 1990. IEEE.

[5] Caglan M. Aras, Douglas S. Reeves, and Ren C. Luo. Low latency, high acceptance real-time communications on wide area networks. Technical Report TR-92-010, ECE Department, North Carolina State University, Raleigh, NC, December 1992.

[6] G. A. Awater and F. C. Schoute. Performance improvement of fast packet switching by LDOLL queueing. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 562–568 (4C.4), Florence, Italy, May 1992. IEEE.

[7] Jaime Jungok Bae and Tatsuya Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, February 1991.

[8] Anindo Banerjea and Srinivasan Keshav. Queueing delays in rate-controlled networks. Technical Report TR-92-015, International Computer Science Institute, Berkeley, CA, March 1992.

[9] Giulio Barberis, Neviano Dal Degan, and Fulvio Rusina. Vocoded speech through a packet switched network. In *Conference Record of the International Conference on Communications (ICC)*, pages 921–926 (29.3), Chicago, IL, June 1985. IEEE.

[10] Walter R. Beam. *Command, control and communication systems engineering.* McGraw-Hill, NY, 1989.

[11] Nicola Bléfari-Melazzi. Performance analysis of a cascade of packet multiplexers loaded with correlated inputs. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 958–963 (28.05), Orlando, FL, December 1992. IEEE.

[12] Pierre Boyer, Fabrice M. Guillemin, Michel J. Servel, and Jean-Pierre Coudreuse. Spacing cells protects and enhances utilization of ATM network links. *IEEE Network*, 6(5):38–49, September 1992.

[13] Paul T. Brady. A technique for investigating on-off patterns of speech. *Bell System Technical Journal*, 44(1):1 – 22, January 1965.

[14] Paul T. Brady. A statistical analysis of on-off patterns in 16 conversations. *Bell System Technical Journal*, 47(1):73–91, January 1968.

[15] Paul T. Brady. A model for generating on-off speech patterns in two-way conversation. *Bell System Technical Journal*, 48(9):2445–2472, September 1969.

[16] Frank M. Brochin and John B. Thomas. Voice transmission in packet switching networks: a model for queueing analysis. In *26th Annual Allerton Conference on Communication, Control and Computing*, pages 1001–1004, Monticello, IL, September 1988.

[17] Randy Brown. Calendar queues: A fast O(1) priority queue implementation for the simulation event set problem. *Communications of the ACM*, 31(10):1220–1227, October 1988.

[18] Stephen Casner and Stephen Deering. First IETF Internet audiocast. *ACM Computer Communication Review*, 22(3):92–97, July 1992.

[19] Cheng-Shang Chang. Stability, queue length and delay, part I: Deterministic queueing networks. Research Report RC 17708 (#77962), IBM Research Division, Yorktown Heights, NY, February 1992.

[20] Cheng-Shang Chang. Stability, queue length and delay, part II: Stochastic queueing networks. Research Report RC 17709 (#77963), IBM Research Division, Yorktown Heights, NY, February 1992.

[21] Jonathan H. Chao. A novel architecture for queue management in the ATM network. *IEEE Journal on Selected Areas in Communications*, 9(7):1110–1118, September 1991.

[22] Hin Soon Chin, John W. Goodge, Roy Griffiths, and David J. Parish. Statistics of video signals for viewphone-type pictures. *IEEE Journal on Selected Areas in Communications*, 7(5):826–832, June 1992.

[23] Renu Chipalkatti, James F. Kurose, and Don Towsley. Scheduling policies for real-time and non-real time traffic in a statistical multiplexer. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 774–783, Ottawa, Canada, April 1989. IEEE.

[24] I. Cidon and Y. Ofek. Metaring – a full duplex ring with fairness and spatial reuse. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 969–981, San Francisco, CA, 1990.

[25] David D. Clark, Scott Shenker, and Lixia Zhang. Supporting real-time applications in an integrated services packet network: architecture and mechanism. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 14–26, Baltimore, MD, August 1992. ACM.

[26] Jean-Yves Cochennec, Pierre Adam, and Thierry Houdoin. Asynchronous time-division networks: Terminal synchronization for video and sound signals. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 791–794 (25.6), New Orleans, LA, December 1985. IEEE.

[27] Dan Cohen. A protocol for packet-switching voice communication. *Computer Networks*, 2:320–331, September/October 1978.

[28] Rene Leonardo Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[29] Rene Leonardo Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[30] Luiz A. DaSilva, David W. Petr, and Victor S. Frost. A class-oriented replacement technique for lost speech packets. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 1098–1105, Ottawa, Canada, April 1989. IEEE.

[31] Martin De Prycker, Marc Ryckebusch, and Peter Barri. Terminal synchronization in asynchronous networks. In *Conference Record of the International Conference on Communications (ICC)*, pages 800–807, Seattle, WA, June 1987. IEEE.

[32] Luca Delgrossi, Christian Halstrick, Ralf Guido Herrtwich, and Heinrich Stüttgen. HeiTP: a transport protocol for ST-II. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 1369–1373 (40.02), Orlando, FL, December 1992. IEEE.

[33] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 1–12, Austin, TX, September 1989. ACM.

[34] Sylvie Dupuy, Wassim Tawbi, and Eric Horlait. Protocols for high-speed multimedia communications networks. *Computer Communications*, 15(6):349–358, July/August 1992.

[35] Robert E. Easton, P. T. Hutchison, Richard W. Kolor, Richard C. Mondello, and Richard W. Muise. TASI-E communications system. *IEEE Transactions on Communications*, COM-30(4):803–807, April 1982.

[36] A. E. Eckberg. B-ISDN/ATM traffic and congestion control. *IEEE Network*, 6(5):28–37, September 1992.

[37] Anwar Elwalid and Debasis Mitra. Effective bandwidth of general markovian traffic sources and admission control of high-speed networks. *IEEE/ACM Transactions on Networking*, 1(3 (to appear)), June 1993.

[38] Domenico Ferrari. The Tenet group, October 1992.

[39] Domenico Ferrari, Anindo Banerjea, and Hui Zhang. Network support for multimedia — a discussion of the Tenet approach. Technical Report TR-92-072, Computer Science Division, University of California at Berkeley, Berkeley, CA, November 1992.

[40] Domenico Ferrari and Dinesh C. Verma. Scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[41] Bernard Gold. Digital speech networks. *Proceedings of the IEEE*, 65(12):1636–1658, December 1977.

[42] Elliot M. Gold. The conferencing market explodes. *Networking Management*, 10(6):40–48, May 1992.

[43] S. Jamaloddin Golestani. A framing strategy for congestion management. *IEEE Journal on Selected Areas in Communications*, 9(7):1064–1077, September 1991.

[44] Albert G. Greenberg and Neal Madras. How fair is fair queueing? *Journal of the ACM*, 39(3):568–598, July 1992.

[45] Davide Grillo. Interactive voice application handling in wide-area packet-switched networks. In Minoru Akiyama, editor, *Proceedings of the 11th International Teletraffic Congress (ITC)*, pages 1054–1060, Kyoto, Japan, September 1985. North-Holland.

[46] Roch Guérin, Hamid Ahmadi, and Mahmoud Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, September 1991.

[47] Fabrice Guillemin, Pierre Boyer, Alain Dupuis, and Luc Romoeuf. Peak rate enforcement in ATM networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 753–758 (6A.1), Florence, Italy, May 1992. IEEE.

[48] H. W. Gustafson. Model for the analysis of talkspurt and silence durations in conversational interaction. In *Proceedings of the 77th Annual Convention of the American Psychology Association*, volume 44, pages 43–44. APA, 1969.

[49] Rainer Händel and Manfred N. Huber. *Integrated broadband networks: An introduction to ATM-based networks.* Addison-Wesley, Wokingham, England, 1991.

[50] Wm. A. Hargreaves. A model for speech unit duration. *Language and Speech*, 3(3):164–173, July – Sept 1960.

[51] B.G. Haskell. Buffer and channel sharing by several interframe picturephone coders. *Bell System Technical Journal*, 51(1):261–289, January 1972.

[52] Gérard Hebuterne and Annie Gravey. A space priority queueing mechanism for multiplexing ATM channels. *Computer Networks and ISDN Systems*, 20(1–5):37–43, December 1990. ITC Specialist Seminar, 25–29 September 1989, Adelaide, Australia.

[53] Daniel P. Heymann, Ali Tabatabei, and T. V. Lakshman. Statistical analysis and simulation study of video teleconference traffic in ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2(1):49–59, March 1992.

[54] Nian-Chyi Huang. An integrated voice/data network architecture using virtual circuits. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 542–546 (17.6), New Orleans, LA, December 1985. IEEE.

[55] Shan-shan Huang. Modeling and analysis for packet video. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, volume 2, pages 881–885 (25.2), Dallas, TX, November 1989. IEEE.

[56] Joseph Y. Hui. Resource allocation for broadband networks. *IEEE Journal on Selected Areas in Communications*, 6(9):1598–1608, December 1988.

[57] Ren-Hung Hwang, James F. Kurose, and Don Towsley. The effect of processing delay and QOS requirements in high-speed networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 160–169 (2A.3), Florence, Italy, May 1992. IEEE.

[58] Jay Hyman, Aurel A. Lazar, and Giovanni Pacifici. Joint scheduling and admission control for ATS-based switching nodes. In *SIGCOMM Symposium on Communications Architectures and*

Protocols, pages 223–234, Baltimore, MD, August 1992. ACM and IEEE. also in *Computer Communications Review*, Vol. 22 (4).

[59] Jay M. Hyman, Aurel A. Lazar, and Giovanni Pacifici. Real-time scheduling with quality of service constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, September 1991.

[60] Joseph Jaffe, Louis Cassotta, and Stanley Feldstein. Markovian model of time patterns of speech. *Science*, 144:884–886, 1964.

[61] Sugih Jamin, Scott Shenker, Lixia Zhang, and David 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, San Diego, CA, November 1992. IEEE Computer and Communications Societies.

[62] C. R. Kalmanek, H. Kanakia, and Srinivasan Keshav. Rate controlled servers for very high-speed networks. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 12–20 (300.3), San Diego, CA, December 1990. IEEE.

[63] Dilip D. Kandlur, Kang G. Shin, and Domenico Ferrari. Real-time communication in multi-hop networks. In *11th International Conference on Distributed Computing Systems (DCS)*, pages 300–307, Arlington, TX, 1991.

[64] V. R. Karanam, K. Sriram, and Duane O. Bowker. Performance evaluation of variable-bit-rate voice in packet-switched networks. *AT&T Technical Journal*, pages 57–71, September/October 1988.

[65] Shinji Kawaguchi, Mitsuo Tsujikado, Yutaka Ueda, and Kenichiro Hosoda. Quality controlled variable rate coding based on constant error criterion. In *Proceedings of Visicom '90: Third International Workshop on Packet Video*, page C4, Morristown, NJ, March 1990. IEEE.

[66] Srinivasan Keshav. On the efficient implementation of fair queueing. *Journal of Internetworking Research and Experience*, 2:157–173, September 1991.

[67] Hans Kröner, Gérard Hébuterne, Pierre Boyer, and Annie Gravey. Priority management in ATM switching nodes. *IEEE Journal on Selected Areas in Communications*, 9(3):418–427, April 1991.

[68] Kouji Kubota, Masayuki Murata, Hideo Miyahara, and Yuji Oie. Congestion control for bursty video traffic in ATM networks. *Electronics and Communications in Japan, Part I*, 75(4):13–19, April 1992.

[69] H. T. Kung. Gigabit local area networks: A systems perspective. *IEEE Communications Magazine*, 30(4):79–89, April 1992.

[70] James Kurose. Open issues and challenges in providing quality of service guarantees in high speed networks. *ACM Computer Communication Review*, 23(1):6–15, January 1993.

[71] James F. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Sigmetrics 1992*, pages 128–139, New Port, RI, June 1992. ACM.

[72] Chin-Tau Lea and Anwar Alyatama. Bandwidth quantization in the broadband ISDN. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 21–29 (1A.3), Florence, Italy, May 1992. IEEE.

[73] Duan-Shin Lee, Benjamin Melamed, Amy R. Reibman, and Bhaskar Sengupta. TES modeling for analysis of a video multiplexer. *Performance Evaluation*, 16(1–3):21–34, 1992.

[74] Cheng-Chew Lim, Li jun Yao, and Wei Zhao. A comparative study of three token ring protocols for real-time communicati ons. In *11th Conference on Distributed Computing Systems (DCS)*, pages 308–317, Arlington, TX, 1991.

[75] J.O. Limb. Buffering of data generated by the coding of moving images. *Bell System Technical Journal*, 51(1):239–259, January 1972.

[76] Thomas D. C. Little and Arif Ghafoor. Multimedia synchronization protocols for broadband integrated services. *IEEE Journal on Selected Areas in Communications*, 9(9):1368–1381, December 1991.

[77] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the Association of Computing Machinery*, 20(1):46–61, January 1973.

[78] Carlyn M. Lowery. Protocols for providing performance guarantees in a packet-switching internet. Technical Report TR-91-002, Computer Science Division, University of California at Berkeley, Berkeley, CA, January 1991.

[79] Basil Maglaris, Dimitris Anastassiou, Prodip Sen, Gunnar Karlsson, and John D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications*, 36(7):834–844, July 1988.

[80] N. Malcolm and W. Zhao. Advances in hard real-time communications with local area networks. In *17th IEEE Conference on Local Computer Networks*, Minneapolis, MN, September 13-16 1990. IEEE.

[81] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski. TES-based traffic modeling for performance evaluation of integrated networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 75–84 (1C.1), Florence, Italy, May 1992. IEEE.

[82] Daniel Minoli. Issues in packet voice communications. *Proceedings of the Institution of Electrical Engineers*, 126(8):729–740, August 1979.

[83] Daniel Minoli. Optimal packet length for packet voice communication. *IEEE Transactions on Communications*, COM-27(3):607–611, March 1979.

[84] Hiromi Nagabuchi, Akira Takahashi, and Nobuhiko Kitawaki. Speech quality degraded by cell loss in ATM networks. *NTT Review*, 4(4):45–51, July 1992.

[85] Ramesh Nagarajan, James F. Kurose, and Don Towsley. Approximation techniques for computing packet loss in finite-buffered voice multiplexers. *IEEE Journal on Selected Areas in Communications*, 9(3):368–377, April 1991.

[86] Ramesh Nagarajan, James F. Kurose, and Don Towsley. Local allocation of end-to-end quality-of-service measures in high-speed networks. In *IFIP International Workshop on Modeling of ATM Networks*, pages 2.2.1 – 2.2.29, Le Martinique, French Carribean, January 1993. North Holland.

[87] Ryohei Nakayama, Tomonori Shino, and Kazuya Arino. High-speed packet multiplexing architecture for multimedia communications. *NTT Review*, 4(4):75–80, July 1992.

[88] Patrik Nises and Joakim Wettby. Phonetalk. Master's thesis, Royal Institute of Technology, Stockholm, Sweden, December 1990.

[89] Mitsuru Nomura, Tetsurou Fujii, and Naohisa Ohta. Basic characteristics of variable rate video coding in ATM environments. *IEEE Journal on Selected Areas in Communications*, 7(1):752–760, June 1989.

[90] Ilkka Norros, James W. Roberts, Alain Simonian, and Jorma T. Virtamo. The superposition of variable bit rate sources in an ATM multiplexer. *IEEE Journal on Selected Areas in Communications*, 9(3):378–387, April 1991.

[91] A. C. Norwine and O. J. Murphy. Characteristic time intervals in telephone conversations. *Bell System Technical Journal*, 17(4):281–291, April 1938.

[92] Yoshihiro Ohba, Masayuki Murata, and Hideo Miyahara. Analysis of interdeparture processes for bursty traffic in ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):468–476, April 1991.

[93] Pramod Pancha and Magda El Zarki. A look at the MPEG video coding standard for variable bit rate transmission. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, page (1A.2), Florence, Italy, May 1992. IEEE.

[94] Abhay K. Parekh and Robert G. Gallagher. A generalized processor sharing approach to flow control in integrated serv ices networks:the multiple node case. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 521–530, San Francisco, CA, 1993.

[95] Colin Parris, Hui Zhang, and Domenico Ferrari. A mechanism for dynamic re-routing of real-time channels. Technical Report TR-92-053, International Computer Science Institute, Berkeley, CA, March 1992.

[96] Craig Partridge and Stephen Pink. An implementation of the revised Internet Stream Protocol (ST-2). In *Second International Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, Germany, November 1991. ACM Sigcomm.

[97] David W. Petr and Victor S. Frost. Optimal packet discarding: An ATM-oriented analysis model and initial results. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 537–542, San Francisco, CA, June 1990. IEEE.

[98] David W. Petr and Victor S. Frost. Nested threshold cell discarding for ATM overload control: optimization under cell loss constraints. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 1403–1412 (12A.4), Bal Harbour, FL, April 1991. IEEE.

[99] Sridhar Pingali and James F. Kurose. On scheduling two classes of real time traffic with identical deadlines. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 460–465, Phoenix, AZ, December 1991. IEEE.

[100] G. Ramamurthy and B. Sengupta. Modeling and analysis of a variable bit rate video multiplexer. In *International Teletraffic Congress, Seventh Specialist Seminar*, pages 8.4.1 – 8.4.8, Morristown, NJ, October 1990. ITC.

[101] G. Ramamurthy and B. Sengupta. Modeling and analysis of a variable bit rate video multiplexer. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 817–827 (6C.1), Florence, Italy, May 1992. IEEE.

[102] J. W. Roberts, J. Guibert, and A. Simonian. Network performance considerations in the design of a VBR codec. In J. W. Cohen and Charles D. Pack, editors, *Queueing, Performance and Control in ATM — Proceedings of the Workshop at the 13th International Teletraffic Congress (ITC)*, pages 77–82, Copenhagen, Denmark, June 1991. North-Holland. Volume 15 of the North Holland Studies in Telecommunication.

[103] Ramon M. Rodriguez-Dagnino, Masoud R. K. Khansari, and Alberto Leon-Garcia. Prediction of bit-rate sequences of encoded video signals. *IEEE Journal on Selected Areas in Communications*, 9(4):305–314, April 1991.

[104] Henning Schulzrinne. RTP: The real-time transport protocol. In *MCNC 2nd Packet Video Workshop*, volume 2, Research Triangle Park, NC, December 1992.

[105] Henning Schulzrinne. Voice communication across the internet: A network voice terminal. Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, MA, July 1992.

[106] Henning Schulzrinne, James F. Kurose, and Don Towsley. Congestion control for real-time traffic in high-speed networks. Technical Report TR 89-92, Department of Computer and Information Science, University of MA, Amherst, MA, 1989.

[107] Henning Schulzrinne, James F. Kurose, and Don Towsley. Congestion control for real-time traffic in high-speed networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 543–550, San Francisco, CA, June 1990.

[108] Henning G. Schulzrinne. *Reducing and characterizing packet loss for high-speed computer networks with real-time services.* PhD thesis, University of Massachusetts, Amherst, MA, May 1993.

[109] A. J. Seyler. Probability distributions of television frame differences. *Proceedings of the Institution of Radio and Electronics Engineers (IREE) Australia*, 26(11):355–366, November 1965.

[110] K. G. Shin and Q. Zheng. Mixed time-constrained and non-time-constrained communications in local are a networks. *IEEE Transactions on Communications*, in press, October 1993.

[111] K. Sriram. Methodologies for bandwidth allocation, transmission scheduling, and congestion avoidance in broadband ATM networks. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 1545–1551 (44.08), Orlando, FL, December 1992. IEEE.

[112] K. Sriram and David M. Lucantoni. Traffic smoothing effects of bit dropping in a packet voice multiplexer. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 759–769, New Orleans, LA, March 1988. IEEE.

[113] Kotikalapudi Sriram and David M. Lucantoni. Traffic smoothing effects of bit dropping in a packet voice multiplexer. *IEEE Transactions on Communications*, 37(7):703–712, July 1989.

[114] John A. Starkwheather. Vocal behaviour: the duration of speech units. *Language and Speech*, 2(3):146–153, July–September 1959.

[115] Jay K. Strosnider, Tom Marchok, and John. Lehochzky. Advanced real-time scheduling using the IEEE802.5 token ring. In *Proceedings of the 1988 Real-Time Systems Symposium*, pages 42–52, Huntsville, AL, December 6-8 1988.

[116] Study Group XVIII, CCITT (International Telegraph and Telephone Consultative Committee). Study group XVIII - report R 34, June 1990.

[117] Claudio Topolcic, Stephen Casner, Charles Lynn, Jr., Philippe Park, and Kenneth Schroder. Experimental internet stream protocol, version 2 (ST-II). Network Working Group Request for Comments RFC 1190, BBN Systems and Technologies, October 1990.

[118] Jonathan S. Turner. New directions in communications (or which way to the information age?). *IEEE Communications Magazine*, 24(10):8–15, October 1986.

[119] Andre M. van Tilborg and Gary M. Koob. *Foundations of Real-Time Computing: Scheduling and Resource Management.* Kluwer Academic Publishers, Boston/Dordrecht/London, 1991.

[120] Willem Verbiest and Luc Pinnoo. A variable bit rate codec for asynchronous transfer mode networks. *IEEE Journal on Selected Areas in Communications*, 7(5):761–770, June 1989.

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

[122] Marcel Verzeano. Time-patterns of speech in normal subjects. *Journal of Speech and Hearing Disorders*, 15(3):197–201, September 1950.

[123] Bernd Wolfinger and Mark Moran. A continuous media data transport service and protocol for real-time communication in high-speed networks. In *Proceedings of the 2nd International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 171–182, Heidelberg, Germany, November 1991.

[124] Gillian M. Woodruff and Rungroj Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE Journal on Selected Areas in Communications*, 8(3):437–446, April 1990.

[125] O. Yaron and M. Sidi. Calculating performance bounds in communication networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, CA, March 1993. IEEE.

[126] Yohtaro Yatsuzuka. Highly sensitive speech detector and high-speed voiceband data discriminator in DSI-ADPCM systems. *IEEE Transactions on Communications*, COM-30(4):739–750, April 1982.

[127] Ferit Yegenoglu, Bijan Jabbari, and Ya-Qin Zhang. Modeling of motion classified VBR video codecs. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 105–109 (1C.4), Florence, Italy, May 1992. IEEE.

[128] J. Zdepski, K. Joseph, and D. Raychaudhuri. Packet transport of VBR interframe DCT compressed digital video on a CSMA/CD LAN. In *Proceedings of the Conference on Global Communications (GLOBECOM)*, pages 886–892 (25.3), CA, December 1989. IEEE.

[129] Hui Zhang and Domenico Ferrari. Rate-controlled static-priority queueing. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 1, pages 227–236, San Francisco, CA, 1993.

[130] Hui Zhang and Srinivasan Keshav. Comparison of rate-based service disciplines. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 113–121, Zurich, Switzerland, September 1991. ACM.

[131] Qin Zheng and Kang G. Shin. Fault-tolerant real-time communication in distributed computing systems. In *Proc. 22nd Annual International Symposium on Fault-Tolerant Computing*, pages 86–93, Boston, MA, 1992.

[132] Qin Zheng and Kang G. Shin. Real-time communication in local area networks. In *Proc. of 17th Conf. on Local Computer Networks*, pages 416–425, Minneapolis, MN, 1992.

[133] Qin Zheng and Kang G. Shin. On the ability of establishing real-time channels in point-to-point packet switched networks. *IEEE Transactions on Communications*, to appear 1993.