A Scheduling Discipline and Admission Control Policy for Xunet 2

Huzur Saran*, Srinivasan Keshav, Charles R. Kalmanek

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974 USA
email: hsaran@cse.iitd.ernet.in, {keshav, crk}@research.att.com

Abstract: Xunet 2 is a collaborative research program with a goal of understanding the fundamental issues in the performance of ATM networks. These networks are expected to carry a mixture of constant bit rate traffic, variable bit rate traffic and computer traffic spanning a wide range of performance requirements. This paper describes these service requirements and matches them with performance guarantees that can be provided by the scheduling discipline supported by an experimental ATM switch. The scheduler supports per-virtual-circuit queueing and several priorities of round robin service in order to segregate real-time and non-real-time traffic and provide fair sharing for bursty computer traffic. Detailed simulations show that real-time traffic can be efficiently integrated with non-real-time traffic using appropriate call admission policies and enhancements to traditional round robin scheduling. While the present study is focused on providing quality of service guarantees in the Xunet 2 network, the design of the scheduler and the call admission policies are relevant to ATM networks in general.

Keywords: Quality of Service, Scheduling, Admission Control, Xunet 2, ATM

1. Introduction

This paper considers the expected performance requirements for traffic on future ATM networks and matches them with performance guarantees that can be provided by the scheduling disciplines supported by an experimental ATM switch. We first address the requirements of video, audio, data and other traffic that we expect will be carried by a future ATM-based telecommunications infrastructure (Section 2). We motivate and describe in detail the capabilities of the hardware scheduler in our switch (Section 3) and we design a scheduling discipline and call admission policy, consistent with the hardware scheduler, that meets the service requirements (Sections 4 and 5). Finally, we use simulations to verify that the performance objectives can be met (Sections 6 and 7). While the purpose of the present study is to provide good support for mixed traffic on the Xunet 2 experimental network, the design of the scheduler and the call admission policies are relevant to traffic management schemes for ATM networks in general.

This research was conducted as part of Xunet 2: a research collaboration involving AT&T and a number of universities and government laboratories. A goal of Xunet 2 is to study the fundamental issues in the performance of ATM networks carrying traffic with widely varying performance requirements. The Xunet 2 collaboration makes use of a wide-area testbed network of ATM switches connected by 45 Megabits/sec links. This network also supports the BLANCA collaboration in the Gigabit Network project.

2. Service requirements

Although it is difficult to predict the traffic carried by future high speed networks, it is widely believed that the following types of traffic will be important. A service architecture for ATM networks must therefore support these traffic types.

- *Constant Bit Rate (CBR) Traffic*: ATM networks will be used to emulate synchronous circuits, for example, to carry video encoded at 384 Kilobits/sec (Liou 1991) or 1.5 Megabits/sec, or uncompressed voice at 64 Kilobits/sec. At entry to the network this traffic consists of equi-spaced cells which will be subject to delay jitter due to queueing and scheduling in each ATM switch. The delay jitter will typically be absorbed in a *playback buffer* at the receiver, which therefore needs an estimate

* On leave from Indian Institute of Technology, Delhi, India.

of the amount of delay jitter introduced by the network.

- *Variable Bit Rate (VBR) Video*: Video traffic may well comprise a substantial fraction of the traffic on future ATM networks. Video encoders under development today generate a fixed number of frames per second (normally 30), but the number of bits in each frame can vary depending on the visual complexity in the image or the amount of motion if motion compensation is used. It is often assumed that the video encoder will adjust its coding parameters in order to conform to a *leaky bucket* traffic description agreed upon with the network, which limits the encoder's peak rate, average rate, and burst size. Kanakia et al (Kanakia et al 1993) measured the output of an unconstrained MPEG encoder for different scenes, and found that the average rates vary from 3 to 7 Mbps, and the peak rates from 8 to 12 Mbps.

  As with CBR traffic, VBR traffic requires a playback buffer at the receiver whose size depends on the delay jitter introduced by the network. As a result the delay-jitter should be kept small; a delay jitter comparable to one video frame time (about 30 milliseconds at 30 frames per second) may be acceptable since typical decoders already have a three frame buffer. We refer to both CBR and VBR video traffic as ‘‘real-time’’ since this traffic is sensitive to variations in delay. For interactive applications the total delay may also be an issue.

  Workstation-based multimedia applications may allow a wider latitude in trading off the image quality, bandwidth requirements, and delay tolerance. The traffic generated from such sources may differ significantly from traditional video coders. Since such traffic is still evolving and no widely accepted traffic models are available, we do not consider such traffic further in this paper.

- *Network Control and Telemetry*: Some applications, such as network control and telemetry, require low bandwidth but also low delay. For example, the bandwidth requirements may be no more than 10 Kilobits/second. The network may wish to offer a special service for this low bandwidth, but ‘‘urgent’’ traffic.

- *High and Low Priority Data* : Computer traffic is bursty with a large ratio of peak to average rate (Pawlita 1981; Gusella 1990). The peak rate might be taken to be equal to the speed of the access line, but these sources typically desire to adapt their sending rate to make use of the excess capacity or *Available Bit Rate (ABR)* in the network at the time. In our view, this traffic is not subject to leaky bucket policing, instead the network uses round robin scheduling and a suitable cell discard policy to insure that one user cannot adversely affect the service seen by another user. Sources that seek to maximize throughput will avoid cell loss by adapting their sending rate based on implicit or explicit feedback from the network. The Packet Pair protocol (Keshav 1991) is one example of a rate adaptation scheme.

  Computer traffic does not ordinarily have tight delay requirements, but prefers that the network not lose data since any data that is lost must be retransmitted. We consider the possibility that the network may offer a low priority and a high priority data service. The low priority or ‘‘bulk rate’’ service makes use of capacity that is unused by the higher priority data service. Examples of high priority data traffic are interactive access to remote information servers, interactive file transfer, et cetera. Bulk data transfer is meant for electronic mail, network news, and other such delay-insensitive applications.

3. Xunet 2 Cell Scheduler

The Xunet 2 testbed is comprised of experimental ATM switches interconnected via 45 Megabit/sec transmission lines*. The switches are output-queued: cell scheduling at the outputs is implemented by a queueing engine (Kalmanek et al 1992) that supports per virtual circuit queueing and several priority levels of round robin service as described in detail below. The per virtual circuit queues are implemented in a single statistically shared memory, but no single virtual circuit's queue can consume more memory than the queue

*There are also several transmission lines which operate at 622 Megabits/sec.

length *limit* for that virtual circuit. The implementation allows a different limit to be set for each virtual circuit.

The importance of per virtual circuit queueing and round robin service in data networks has been addressed elsewhere (Morgan 1991), but we briefly motivate the design. Round robin service assures that when the network becomes congested, bandwidth is shared fairly among active users. Provided that sources adapt their sending rate appropriately, no queue will grow very large, and sources that send too fast only increase the length of their own queue. If a source consistently sends too fast, its data will be dropped. In contrast, with first in first out scheduling, a user can consume an arbitrary fraction of the network bandwidth and can cause data sent by well-behaved users to be dropped.

In our implementation, each virtual circuit has a *service class* associated with it that determines its treatment by the scheduler. In addition to ordinary round robin the queueing engine supports several levels of priority, so that virtual circuits in one service class can be given higher priority than those in another service class. We refer to Fig. 1 in order to describe round robin with multiple priorities.

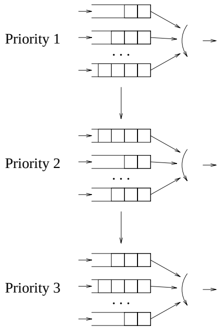

Fig. 1: Multiple priority round-robin service

For each circuit at the highest priority with data waiting to be transmitted, the scheduler serves one cell and then moves on to the next circuit. If no circuit at the highest priority has data to send, the scheduler will serve circuits at the second highest priority, returning at the end of each cell service time to the highest priority circuits if necessary. Circuits at the third highest priority are served when there are no priority 1 or 2 circuits to be served, and so on. The round robin scheduler is implemented using a *control queue* for each level of priority. The control queue contains a list of virtual circuit identifiers with data waiting to be transmitted. The scheduler removes a virtual circuit from the head of the control queue when it is served and returns it to the end of the queue if there is still data waiting to be transmitted.

The queueing engine also supports two variations of round robin scheduling, weighted round robin and framed round robin. Weighted round robin allows different virtual circuits to receive different proportions of the available bandwidth by serving each virtual circuit for a number of time slots corresponding to its weight. Our implementation of weighted round robin, described in the next section, is equivalent to a scheduling discipline known as Rate-Proportional Processor Scheduling in which the weight given to a virtual circuit is proportional to the fraction of link capacity it uses. The analysis in (Parekh 1992) gives the worst case bounds on delay and delay jitter. Framed round robin insures that no more than a given number of cells are transmitted during an interval known as the *frame time*. Framed round robin has the effect of "smoothing" the traffic stream arriving at the downstream switch, which affects the amount of buffering needed at that switch. Framed round robin is a subset of a scheduling discipline known as Hierarchical

Round Robin (Kalmanek et al 1990).

In order to support weighted round robin, conceptually each virtual circuit has a weight or *service quantum* associated with it*. When the virtual circuit identifier is removed from the control queue to be served, the circuit can be served for a number of cells equal to the service quantum, or less if fewer cells are waiting. The virtual circuit identifier is then returned to the control queue if more data is waiting. Framed round robin allows a virtual circuit to be served up to some number of cell service times during a frame time. This is implemented with two control queues which are served alternately during successive frames. Once a cell has received its service quantum during a frame, it is appended to the control queue which is not currently being served, insuring that it will not received any further cell services until the succeeding frame.

4. Scheduling discipline and admission control design

In this section, we discuss the design of a scheduling discipline that meets the bandwidth and delay jitter requirements of the different traffic classes described in Section 2. We begin by discussing a priority structure for the traffic classes, then, for each priority level, we describe the scheduling discipline, admission control and policing mechanism. We do not need to explicitly consider bounds on total delay for the real-time CBR or VBR video traffic: an application program is expected to request a large enough bandwidth to meet its end-to-end delay bound.

The network control traffic is most sensitive to delay and uses relatively little bandwidth therefore it is assigned the highest priority. Real-time traffic must be assigned a higher priority than data traffic in order to insure that it receives the bandwidth that it has reserved. Thus, we will clearly need at least three priority levels: for urgent traffic, real-time traffic, and data traffic. Within each priority level, we implement variants of round robin scheduling.

Call admission control for urgent traffic at the highest priority is based on the requested bandwidth and must ensure that the sum of the peak arrival rates over all admitted virtual circuits is less than a few percent of the total trunk capacity. Thus, the probability of a large number of cells arriving on this level in a short duration is negligible and no more than a few cells will accumulate in any queue. The queue length limit can be small and urgent traffic will not significantly affect the performance seen by the lower priority levels. Since this traffic is at the highest priority level, it can be abused by users who send faster than their negotiated rate. To prevent this, urgent traffic is subject to peak-rate policing at the edge of the network. The end-to-end delay for this traffic is only slightly above the propagation delay, thus the admission control need not take into account the end-to-end delay requirements but can use much simpler bandwidth requirements as mentioned above.

Choosing the service discipline for the next priority level, which carries real-time traffic, requires some thought. We assume that leaky bucket admission control at the source constrains the average and peak arrival rates of real-time traffic. With leaky bucket constrained sources, recent research suggests that both weighted round robin (WRR) and framed round robin (FRR) seem to be good choices (Parekh 1992; Zhang and Keshav 1991) since the network is able to provide bandwidth guarantees and bounds on delay jitter using either discipline. However, while both disciplines have a similar worst-case end-to-end delay bound, FRR requires fewer buffers within the network to avoid due to buffer overflow losses than WRR. With the non-work-conserving FRR discipline, bursts admitted by the leaky bucket will be buffered at the first switch in the path, and traffic seen by downstream switches is the smoothed traffic stream produced by the FRR scheduler. Therefore, a buffer of only a few ATM cells at internal switches is sufficient to avoid cell loss due to overflow. On the other hand, the work-conserving WRR discipline allows bursts into the network, and in the worst case, every switch in the path needs have a buffer equal in size or larger than the negotiated leaky bucket size. The tradeoff is that the average queueing delay for FRR will be higher than for WRR, since a lightly loaded WRR server will pass traffic through with small delay, whereas a FRR server always introduces a smoothing delay.

*In the implementation, a weight is associated with each service class rather than each virtual circuit, so we can implement weighted round robin scheduling with the limitation that the number of different weights is limited by the number of available service classes.

Unfortunately, because of the wide range in the bandwidth requirements of real time traffic, it is undesirable to place all the real time traffic at one priority level, whether it be FRR or WRR. There is a tradeoff between the bandwidth granularity and the jitter provided by a FRR or WRR service. Consider a FRR scheduler with a frame time corresponding to *k* cells being serviced on the output trunk. Such a scheduler allows the bandwidth used by that priority level to be divided into at most *k* equal parts, which determines the lowest bandwidth circuit that can be supported. The worst case delay suffered by a circuit at that level is proportional to the frame time since a cell that arrives near the beginning of one frame may not be served until the end of the succeeding frame (if it had already received its service quanta for the frame in which the cell arrived). The tradeoff between bandwidth granularity and jitter is similar for a WRR scheduler.

Therefore, with either a FRR or WRR scheduler, if we wish to serve a wide range of bandwidth requirements, we need to have a large *k* in order to get fine granularity in bandwidth allocation. However, large *k* increases worst case delay for all traffic using that level. It appears that the desired service goals are better met using at least two priority levels, one for high bandwidth traffic (e.g. video) and one for lower bandwidth traffic (e.g. voice). Since the queueing engine allows only one level of framed round robin service, we propose to use a higher priority FRR level and a lower priority WRR level.

We choose this arrangement for several reasons. First, by placing the higher bandwidth traffic in the FRR level, we minimize the use of buffers at internal switches. Second, in order to meet the service requirements of lower priority levels we need to ensure that the higher levels do not use up more than their share of the trunk bandwidth. FRR is guaranteed not to use more bandwidth than its allocation, whereas WRR does not provide this assurance (although the assumption of leaky bucket constrained sources does bound the time between regeneration intervals). Finally, having two different real-time service disciplines running in parallel allows us to compare their relative performance.

The WRR level is guaranteed a minimum bandwidth allocation using a simple scheme. Let the FRR frame be *k* cell intervals (or *slots*) in length. We allocate only $k_1 < k$ slots at the FRR level; the unallocated slots will be used by circuits at lower levels, if any. As a result the FRR level consumes no more than k_1/k fraction of the total bandwidth and circuits at lower levels are guaranteed the remaining bandwidth. Note that the WRR level frame can be made arbitrarily large. We serve $k - k_1$ cells from this frame in round robin order every time the FRR frame of length *k* is served. Thus, if the WRR level has *l* slots, each slot would correspond to $(1 - \frac{k_1}{k})/l$ fraction of the link capacity.

Admission control at both the FRR and WRR levels uses a negotiated leaky bucket traffic descriptor to ensure that links are not overloaded and that delay guarantees are met. We expect applications to size the average rate and token buffer size so that end-to-end delay constraints are met. Admission control thus deals only with checking bandwidth constraints. At the FRR level, the sum of the average rates of the admitted virtual circuits should not exceed some fraction *f* of the trunk capacity. At the WRR level, the sum should not exceed a fraction $1 - f - \varepsilon$, where a fraction ε of the trunk is given to urgent traffic. The queue length limit for the FRR traffic is the leaky bucket size at the first switch, and twice the weight at the internal switches. The queue length limit for WRR traffic is the leaky bucket size at all switches along the path.

Both the FRR and WRR disciplines do not need separate cell level policing for virtual circuits. At the FRR level the framed discipline automatically polices the traffic. At the WRR level any virtual circuit not obeying the negotiated traffic specification cannot affect the performance seen by other WRR circuits, although it can affect the performance of virtual circuits at lower priority levels. Cell level policing may therefore be necessary only in order to ensure that data traffic receives reasonable service from the network.

We propose two priority levels for non-real-time data traffic, corresponding to priority and bulk data transfer. The two levels use work-conserving round robin. If some virtual circuits require relatively more bandwidth, it is possible to use weighted round robin for data traffic as well, giving virtual circuits with higher demands a larger weight. There is no admission control for virtual circuits at these levels, since this traffic makes use of available bandwidth in the network, although the admission control schemes at the higher levels may set aside some bandwidth for data traffic. There is also no explicit policing for data traffic, since the combination of round robin and the queue length limit implicitly police the traffic by sharing bandwidth fairly and punishing users who send too fast. Users are assumed to adapt their sending rate to the rate of the bottleneck link in the network. The queue length limit is set large enough to allow a reasonable deviation above the buffer setpoint used by rate adaption scheme, but small enough so that no source can consume more than a small fraction of the total buffer.

5. Some engineering guidelines

To get a feel for the proposed scheduling policy, we choose a sample set of engineering parameters for a 45 Megabits/sec link carrying a mixture of urgent traffic, CBR video traffic at 384 Kilobits/sec and 1.4 Megabits/sec, CBR voice traffic at 64 Kilobits/sec, VBR traffic, and two priorities of data traffic, labeled BE1 (best effort 1) and BE2 (best effort 2).

- The highest priority level is used for urgent traffic. Call admission control ensures that no more than 2% of the total bandwidth (˜ 1 Megabits/sec) is allocated for this traffic class.
- At a link speed of 45 Megabits/sec, the time to send an ATM cell over the link is approximately 10 microseconds. The frame size for the FRR level is chosen to be 90 slots, so that each slot corresponds to 0.5 Mbps. Call admission ensures that at most 60 slots or 30 Megabits/sec can be reserved for FRR traffic. Thus, the FRR level can accommodate as many as 20 1.5 Megabits/sec CBR video sources. A 384 Kilobits/sec CBR source would reserve one slot and a 1.5 Megabits/sec CBR source would be given a weight of three and would reserve three slots. A cell from a CBR source at the FRR level suffers a maximum delay and delay jitter of roughly 2 * 0.9 = 1.8 milliseconds per switch (Kalmanek et al 1990). Thus, even over a path of 15 switches the delay jitter is no more than 27 milliseconds, which is less than 1 video frame time.
- Of the 90 slots at the FRR level, at most 2 slots (1 Mbps) are used by urgent traffic, and at most 60 by FRR traffic. The residual bandwidth is available for WRR or lower levels, which therefore receive at least 90 - 2 - 60 = 28 slots or 14 Megabits/sec. We divide this bandwidth into 200 slots, where each slot corresponds to (28/90) * 45 Megabits/sec = 70 Kilobits/sec, suitable for 64 Kilobits/sec voice calls. Each virtual circuit at the WRR level is allocated some integral number of slots in order to guarantee bandwidth greater than the desired average rate. By allocating a fixed number of time slots for WRR traffic, with each slot corresponding to a certain average bandwidth, we are able to make use of the worst case bounds on delay and delay jitter for Rate-Proportional Processor Sharing (Parekh 1992).

We have not set aside any bandwidth for data traffic in the above and have not assumed any statistical multiplexing gain that might result if the network were carrying variable bit rate encoded video. Any residual bandwidth not used by the highest priorities would be used by the two best effort levels.

The guidelines described above are summarized in Table 1.

Level #	Type	#slots	Bandwidth/slot	Net bandwidth	Cycle time	
Policing						
1	Urgent	-	-	1Mbps	-	Yes
2	FRR	60	0.5Mbps	30Mbps	0.9ms	No
3	WRR	200	70Kbps	14Mbps	≤6.5ms	No
4	BE1	-	-	Available	-	No
5	BE2	-	-	Available	-	No

Table 1 Engineering guidelines for proposed scheduling discipline

6. Simulation results

The scheduling discipline discussed above was implemented in a network simulator (the REAL simulator developed by one of the authors (Keshav 1988)). Our aim was to understand the behavior of the proposed scheme under heavy load; if the scheme performs satisfactorily at high loads, it will also perform well at lower loads. We studied the performance of urgent, VBR video, voice and data traffic over a network of four switches as shown in Fig. 2.

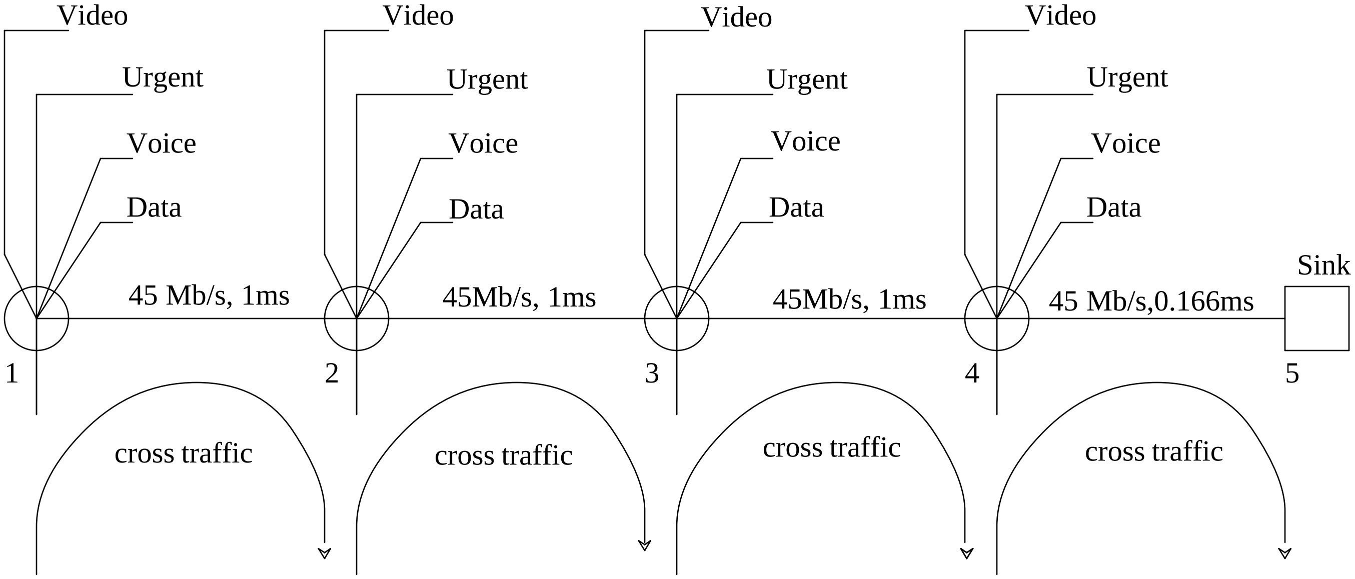

Fig. 2. Simulation topology

The switches are connected by 45 Megabits/sec links with a latency of 1 millisecond while access links to endpoints (sources or sinks of data) are at 45 Megabits/sec with a latency of 0.166 milliseconds. A source that generates each of the four traffic types is connected to each of switches 1, 2, 3 and 4, directing its traffic to a common sink labeled 5. There is enough cross traffic on each link to saturate it, and the cross traffic also goes to the same shared sink. Thus, the amount of cross traffic arriving at a switch decreases as we move closer to the sink.

The urgent stream is modeled by a traffic source with inter-arrival times chosen from a uniform distribution whose mid-point corresponds to an average rate of 10 Kilobits/sec, and whose left-end point corresponds to a peak rate of 45 Megabits/sec. We call this type of traffic ‘‘random rate-controlled’’. Each video stream is generated using the trace of an MPEG coder coding standard video scenes (Kanakia et al 1993). The MPEG coder generates a variable length frame every 33 milliseconds which is transmitted to the switch at the peak bandwidth of 45 Megabits/sec. We allocate a fixed bandwidth of 5.5 Megabits/sec to each video stream and place it at the FRR level. In practice, an MPEG coder could use feedback from an emulation of its leaky bucket to control its burst length and ensure that the network buffers were not overrun. However, since we were using traces, the simulations use an unconstrained MPEG codec (the implications are discussed further in the discussion of simulation results). Each voice stream is generated at a constant bit rate of 64 Kilobits/sec and is placed at the WRR level, where it is allocated a rate of 70 Kbps. The data streams which enter the network at switches 1 and 4 use an infinite data source flow controlled by the packet-pair flow control protocol (Keshav 1991). The data streams which enter the network at switches 2 and 3 use a random rate-controlled source with 1 Megabits/sec average rate and 45 Megabits/sec peak rate. All data traffic is placed at one best effort level (in our simulation we did not use the second best effort level). Finally, adequate random-rate cross-traffic was chosen at each level to load the lines to 100% capacity. Our results show that:

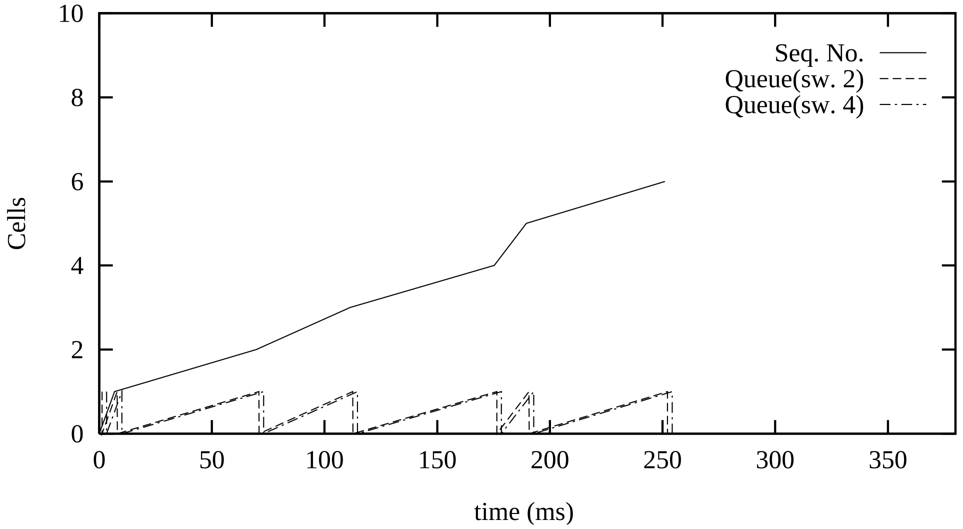

Fig. 3a Urgent traffic at switch 1: Sequence number and queue length

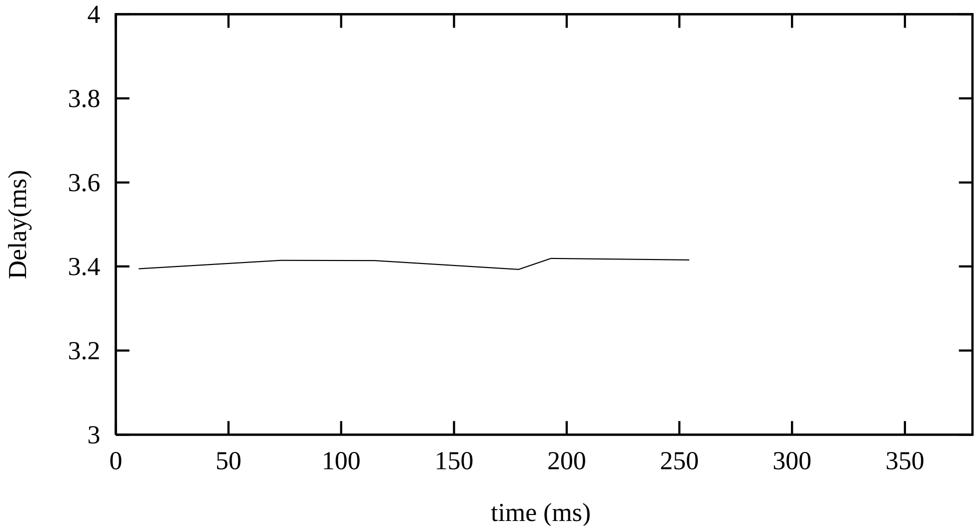

Fig. 3b Urgent traffic at switch 1: End-to-end delay

- *Urgent traffic gets an end-to-end delay very close to the propagation delay.* Figs. 3a and 3b show measurements for priority traffic that enters the network at switch 1 (queue sizes are monitored at switches 2 and 4). Note that there is no long-term queueing and each packet is served almost immediately. The end-to-end delay is essentially equal to the propagation and cell transmission delay of

3.33 milliseconds.

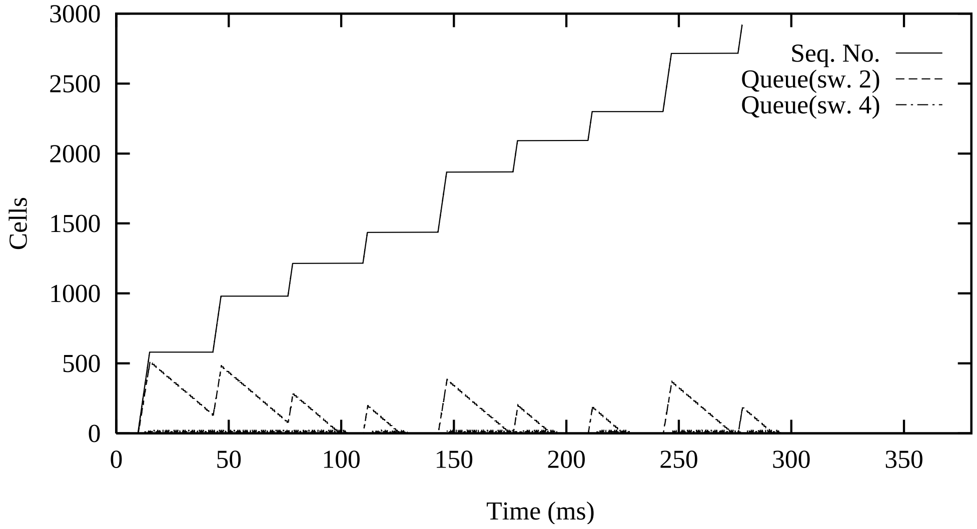

Fig.4a MPEG video at switch 2: Sequence number and queue length

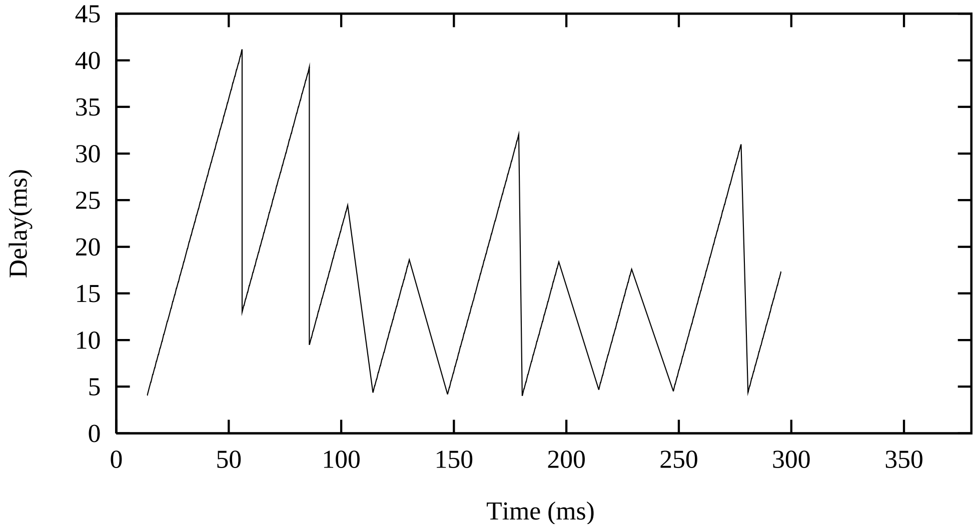

Fig.4b MPEG video at switch 2: End-to-end delay

- *The MPEG video traffic receives a large delay due to queueing at the first switch, and negligible queueing delay at subsequent switches.* In Figs. 4a and 4b we plot the sequence number trace, queue size trace and end-to-end delay of the MPEG video traffic that originates at switch 2 (the queue sizes are monitored at switches 2 and 4). The MPEG coder we simulated acts as an ON-OFF source, so queues build up in the first switch during the ON period (when a frame is received) and drain out at the allocated service rate in the FRR scheduler. This necessarily leads to a delay of roughly one video frame for the cells near the end of the frame. The delay jitter is also comparable to the video frame time. The small additional jitter due to the FRR framing is so small that it is invisible in the plot. If cells were emitted by the MPEG source in a smoothed fashion, as they become available from the coder, the large delay at the first switch would be avoided. This is related to the next point.

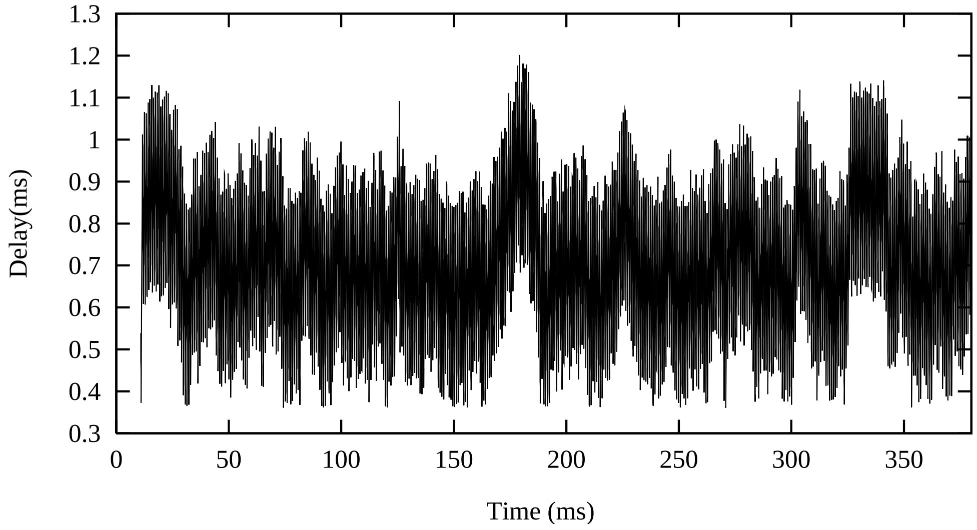

Fig. 5 Random-rate traffic: End-to-end delay

- *The end-to-end delay experienced by FRR traffic depends significantly on the traffic generation pattern of the input source.* If we replace the MPEG coded video source at switch 4 by a random rate source that obeys the peak and average rate behavior, the average delay is close to one frame time per switch (Fig. 5). The observed worst case end-to-end delay is 1.2 milliseconds, of which 0.37 milliseconds are due to propagation and transmission delays. The worst case queueing delay is thus ˜0.8 milliseconds, which is only about 45% of the computed worst case of 1.8 milliseconds.

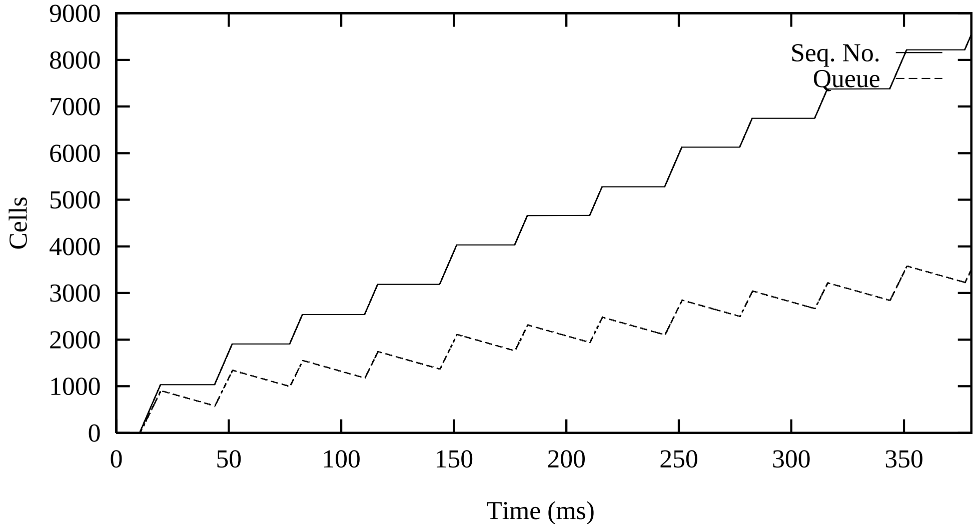

Fig. 6 MPEG video: Sequence number and queue length

- *There is no way to estimate* a priori *the average rate of an MPEG source..* The average rate of a

MPEG encoded video stream can vary from 3 to 7 Megabits/sec as we noted earlier, depending on the image complexity and amount of motion. Unfortunately, if the average rate reserved at call setup is underestimated, queues will build up indefinitely until data is dropped. Fig. 6 shows this behavior for a different image sequence that is allocated the same 5.5 Megabits/sec as was used for the image sequence in Fig. 4. Thus, to use unconstrained MPEG source in conjunction with bandwidth reservation schemes such as FRR or WRR, it will be necessary to reserve near the peak rate, casting serious doubt on the possibility of achieving a good statistical multiplexing gain. While it is possible for the codec to reduce the image quality by adapting to an emulation of its leaky bucket policing, choosing the leaky bucket parameters is an open problem. If the parameters are chosen without sufficient care, there may be an unacceptable reduction in quality. Adaption of the coding rate to network feedback is an option that partly solves this problem, and is addressed in more detail in (Kanakia et al 1993).

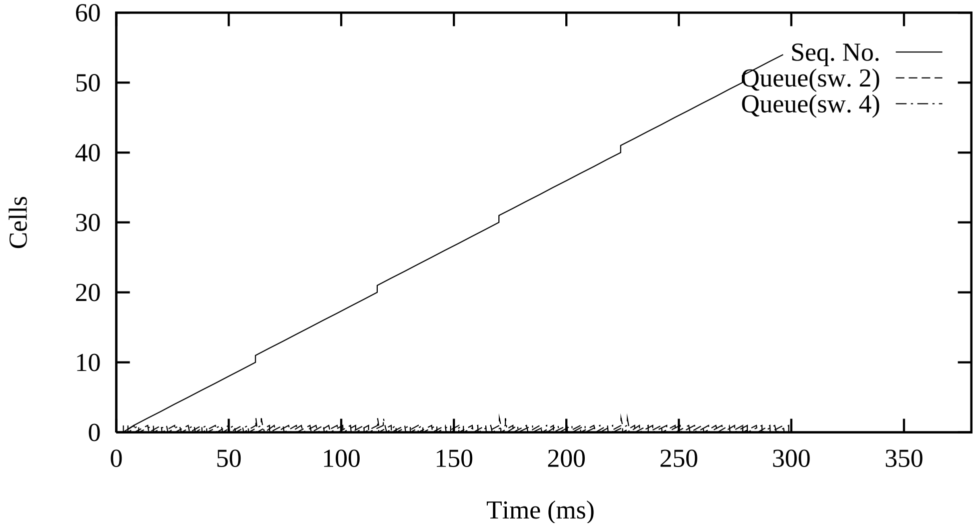

Fig. 7a CBR audio at switch 2: Sequence number and queue length

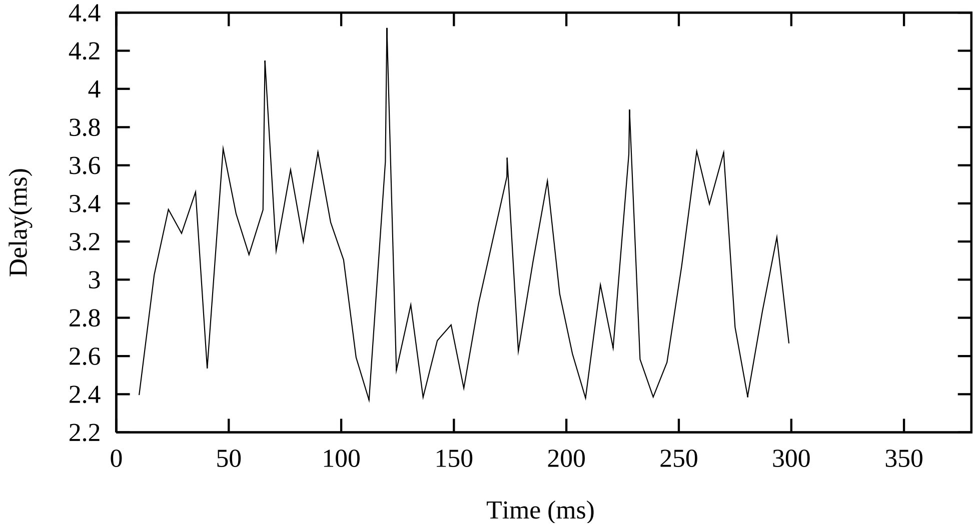

Fig. 7b CBR audio at switch 2: End-to-end delay

- *The WRR traffic receives a per-switch queueing delay well below the computed worst case even when the switch is heavily loaded.* In Figs. 7a and 7b we plot the sequence number queue sizes and end-to-end delay versus time of CBR audio traffic that originates at switch 2. Queue lengths are monitored at switches 2 and 4. The queue size is normally 0 or 1 and there is no queue buildup since the allocated bandwidth of 70 Kilobits/sec is more than arrival rate of 64 Kilobits/sec. The maximum end to end delay is approximately 4.3 milliseconds of which propagation delay and cell transmission times account for 2.37 milliseconds. The remaining delay of 1.93 milliseconds is due to queueing delay. Since WRR traffic is placed in a frame with worst case queueing delay of 6.5 milliseconds (Table 1) per switch, or 19.5 milliseconds end-to-end, the observed worst case is thus only 9.9% of the computed worst case.

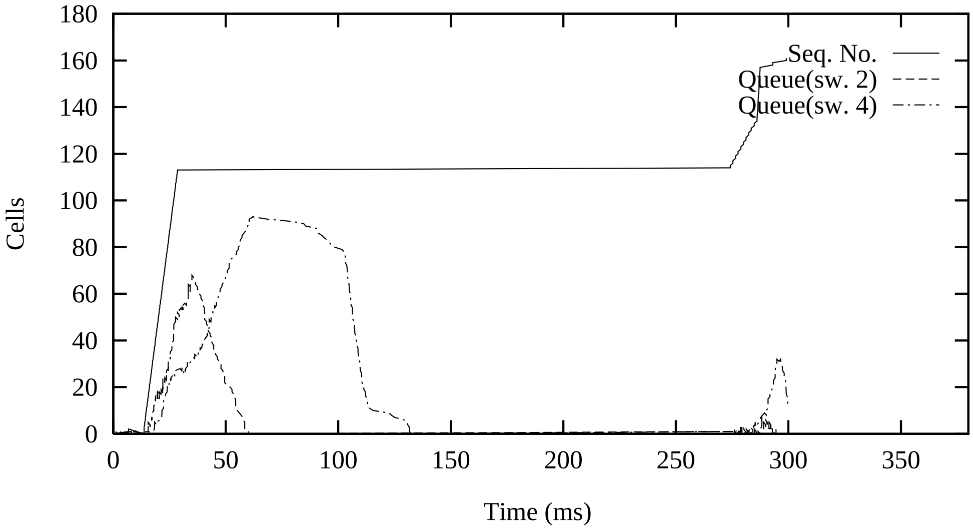

Fig. 8a Best Effort source at switch 1: Sequence number and queue length

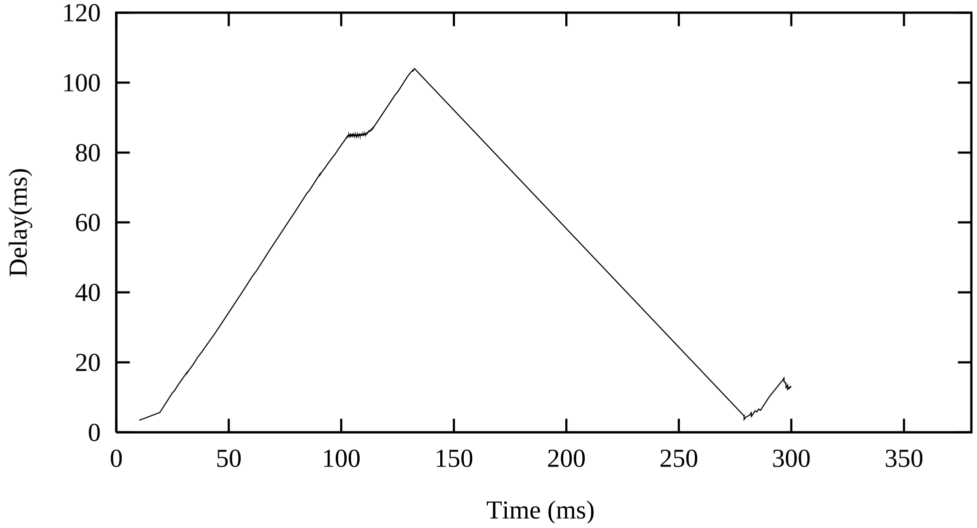

Fig. 8b Best Effort source at switch 1: End-to-end delay

- *Round robin service ensures that the residual bandwidth is allocated fairly over the best-effort traffic.* Best-effort traffic is served only when higher levels do not use their allocated bandwidth. Figs. 8a and 8b plot the behavior of the packet pair flow controlled data source at switch 1. Since the real-time sources start up staggered, there is a large residual bandwidth initially available and the data source gets high throughput. Once all the real-time sources become active, there is very little residual capacity and queues build up at the switches while the packet-pair protocol reacts to this change. The queues flush out when residual bandwidth is available. The end-to-end delay and delay-jitter are, of course, large for this kind of traffic.

7. Conclusions and future work

The simulations show that the scheduling discipline provides the performance requirements of Section 2. The multi-priority structure isolates real-time traffic from non-real-time traffic, and the round-robin structure builds firewalls between sources. The framed discipline allows us to limit queue sizes within the network, while the non-framed disciplines allow us to efficiently statistically multiplex bandwidth.

For unconstrained MPEG sources, we noted a large dependency of end-to-end delay on the smoothing time at the first switch. This has been observed earlier in (Banerjea and Keshav 1993). The real surprise was in the range of variation of the average rate among the MPEG sources. This variation means that bandwidth reservation may be an inefficient way to carry unconstrained MPEG video since the network must reserve resources for the ’worst average-case’ which would be close to the peak-rate. MPEG sources which use local feedback by emulating a leaky bucket have been proposed (Berger et al 1993) along with call admission schemes that attempt to provide a certain cell loss rate based on an *equivalent bandwidth*. These schemes may allow a certain level of statistical multiplexing at the price of a variable image quality. Schemes which use network feedback such as the one in (Kanakia et al 1993) may be more effective at trading off available bandwidth and image quality and may also achieve a higher statistical multiplexing level.

We observe that the actual end-to-end delay seen by traffic served with WRR is far less than the worst case bound computed with the worst case frame delay per switch. Thus, with enough buffers, there is a case to be made for using WRR for serving real-time traffic, particularly if users do not require strict delay-jitter bounds.

As regards future work, we intend to implement the proposed scheduling discipline on Xunet 2, and study the associated call admission and signaling issues. We currently have cameras and video capture hardware attached to a Sparcstation 10 and an Silicon Graphics Indigo on an FDDI ring attached to Xunet 2. We are experimenting with several audio and video tools developed by the Internet community, with the goal of applying the lessons learned to the development of audio/video applications that directly use ATM to provide explicit support for quality of service.

There seem to be two interesting approaches to providing quality of service guarantees for VBR video traffic. In networks that require bandwidth reservation, we have already mentioned the difficulty of picking the appropriate rate. Reservations may be tenable if users have the option of re-negotiating the rate to achieve the desired QOS. The second approach that we intend to explore is the transmission of JPEG and MPEG video with feedback flow control as in (Kanakia et al 1993). We intend to explore the design of a quality of service manager: an application program that coordinates requests to the network for resources according to the user’s dynamic quality of service needs.

8. Acknowledgment

We would like to thank Sam Morgan for many useful discussions.

9. References

Banerjea A, Keshav S (1993) Queueing Delays in Rate Controlled ATM Networks. Proc INFOCOM 1993, San Francisco

Berger AW, Morgan SP, Reibman AR (1993) Statistical Multiplexing of Layered Coded Video. Third Broadband ISDN Technical Workshop, Mandelieu La Napoule, France

Gusella R (1990) Characterizing the Variability of Arrival Processes with Indices of Dispersion. TR-90-051, International Computer Science Institute, Berkeley, California

Kalmanek CR, Kanakia H, Keshav S (1990) Rate Controlled Servers for Very High-Speed Networks. Proc GLOBECOM ’90, San Diego, pp 300.3.1-300.3.9

Kalmanek CR, Morgan SP, Restrick III RC (1992) A High-Performance Queueing Engine for ATM Networks. Proc International Switching Symposium, Japan

Kanakia H, Mishra PP, Reibman AR (1993) An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. Proc ACM SigComm 1993, San Francisco

Keshav S (1988) REAL : A Network Simulator. CSD TR 88/472, University of California, Berkeley

Keshav S (1991) Congestion Control in Computer Networks. Ph.D. dissertation, CSD TR 91/649 University of California, Berkeley

Liou M (1991) Overview of the px46 kbits/s video coding standard, Communications of the ACM 34:4, pp. 59-63.

Morgan SP (1991) Queueing Disciplines and Passive Congestion Control in Byte-Stream Networks IEEE Trans. Comm. 39:7, pp. 1097-1106

Parekh AKJ (1992) A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. Ph.D. dissertation, LIDS-TH-2089, Massachusetts Institute of Technology

Pawlita P (1981) Traffic Measurements in Data Networks, Recent Measurement Results, and Some

Implications. IEEE Trans. Comm. 29:4