QUALITY OF SERVICE IN ATM NETWORKS

Dinesh C. Verma
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
and International Computer Science Institute
Berkeley, California, U.S.A.

Abstract

B-ISDN networks of the future will have to handle traffic with a wide range of traffic characteristics and performance requirements. In view of the high bandwidth of these networks and the relatively large propagation delays involved in wide-area B-ISDN networks, the performance requirements can only be provided by reserving resources to communicating clients at the connection establishment time. However, allocation mechanisms for heterogenous bursty traffic usually result in a rather poor utilization of network resources.

In this paper, we propose a simple admission control criterion that can be used to reserve resources for bursty as well as smooth traffic with delay and loss sensitivities. Our scheme leads to a reasonable utilization of the network bandwidth (about 40%) for delay sensitive traffic with moderate burstiness (peak-to-average bandwidth ratios about 4), even under the worst possible conditions. Actual utilizations can be higher if there is smooth traffic or traffic which is not delay-sensitive. Our scheme uses a well-defined traffic specification scheme which is easy to enforce and verify, and able to accommodate arbitrary degree of burstiness. Extensive simulation experiments failed to show that our admission control criterion are incorrect, in the sense that the quality of service requirements of the traffic were always met, even in the worst case. Moreover, the scheme is simple and feasible at the high speeds required of B-ISDN networks.

This research has been supported in part by AT&T Bell Laboratories, Hitachi, Ltd., the University of California under a MICRO grant, the National Science Foundation under Grant No. CDA-8722788, and the International Computer Science Institute. The views and conclusions in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of any of the sponsoring organizations.

1. Introduction

Computer communications of the near future are likely to have both low delay high-bandwidth capabilities and low delay high-bandwidth requirements. Applications, ranging from file transfers and electronic mail to packetized voice and remote graphical visualization of super-computer results, place diverse and stringent performance requirements on networks. The communication abstractions provided by present computer networks fail to provide any guarantees about the delay, loss rate or delay variation their clients may experience. With the rapid transition into the multi-media age, more and more applications will require such guarantees or some quality of service to be provided by the communication network.

These quality of service requirements can be met by providing communication abstractions that provide guarantees about network performance, like the *real-time channels*[Ferr90] or *flows*[Zhan89]. [1] However, providing these abstractions requires dynamic priority scheduling to be implemented in the switches, i.e. involves scheduling packets in the order of priorities calculated on a per-packet basis (like virtual clock mechanisms of [Zhan89]), which is difficult to implement at high speeds. Thus, we need a scheme which preserves the essence of the real-time channel algorithms, i.e. is able to provide performance guarantees, yet avoids the need for complicated scheduling mechanisms. In this paper, we propose some simple schemes for resource management, which allows us to achieve the same goal.

Quality of service guarantees can be provided either pro-actively or reactively[Lein88]. In reactive quality of service management schemes, the sources are allowed to send data in any fashion they choose. Resources are not allocated to individual connections. However, when the network manager senses that the quality of the service provided by the network is in danger, or likely to be in danger, it takes remedial action to correct the situation, possibly by requesting some sources to send data at a lower rate. However, in wide-area networks with high latencies, the network may not be able to react fast enough. Thus, pro-active techniques involving resource management at call establishment time need to be explored.

Network resource management at call establishment comprises of the following framework. At call setup, a user declares the burstiness characteristics of the call to the network with call control parameters. Based on these parameters, the network manager allocates the necessary resources to the given connection to assure the quality required for cell transfer delay and cell loss. If there are insufficient network resources available, the call will be rejected. The network monitors the cell stream coming from the user to verify that the stream conforms to the parameter values declared at call establishment. It also maintains a scheduling policy which allows more important or delay-sensitive cells to be given priority over the less important cells.

We can identify four crucial components of this resource management architecture, (1) *Specification* : the method by which the burstiness and traffic characteristics of the communicating clients are specified; (2) *Rate Control*: the mechanism used by the network to ensure that this contract is being followed by the clients; (3) *Scheduling policy*: determines which cell is to be transmitted next wherever queueing of cells may occur ; and (4) *Admission Control*: the set of rules which allocates resources and determines if it is possible to accept a new connection. Any resource allocation architecture for wide-area B-ISDN networks must specify these components precisely. To the best of our knowledge, no existing work in the the area of resource allocation for ATM networks [Hui88] [Ohni88] [Prot88] [Hari89] specifies all of these areas precisely.

---

[1] Similar ideas can be found also in [Ande88], [Paru89] and [Come88].

In the next section, we describe the environment assumed for our studies. Section 3 sketches the specification and rate control components of our solution to the resource allocation problem. Section 4 sketches our scheduling and admission control policies. Finally, Section 5 presents some simulation results and conclusions are drawn in the last section.

2. ATM Environment

In an ATM network, all information to be transferred from the source to the destination is packed into fixed size cells, which are identified and switched by means of a label in the cell header. The term asynchronous refers to the fact that cells allocated to the same connection may exhibit an irregular recurrence pattern, as cells are filled according to the actual demand. All communication in an ATM network is connection-oriented, i.e. a connection needs to be established before data transmission can begin. While connections can be duplex in general, we will confine our attention in this paper to simplex (one-way) connections. Duplex connections can be handled as two simplex connections in opposite directions.

The delay through an ATM network is determined by three components (1) propagation delay: which depends on the distance between end-users, is independent of the type of switching node used and by no means controllable; (2) switching delay, composed of a fixed part, implementation dependent, but very low[Yeh87] and (3) a queueing delay. The queueing delay can be controlled by restricting the number of connections through the switch.

We are not considering the delays involved in cell assembly and fragmentation at the edges of the network. The focus of the paper is only on bounding the delays in the network. The work can however be extended to incorporate those delays as well.

The cells lost in an ATM network can be divided into three categories, those lost due to errors in transmission, those lost due to lack of buffers at an ATM switch, and cells lost because they were delayed too much. Since errors in transmission can not be controlled, and the cells lost because of delays can be accounted for by bounding delays, we restrict our attention to guaranteeing a maximum loss rate due to buffer overflows in a switch. In order to provide a bound on the queueing delay and the buffer overflow probability in the switches, a simple model of an ATM switch (described in the next subsection) is used.

As a target solution for broadband ISDN, ATM supports a wide range of voice, data and video services. Narrowband services require low bandwidth or bit rates up to 2 Mbits/second. Examples are audio applications (telephony, hi-fi sound, audio library etc.) and short duration data (teletext, facsimile, telemetry, alarms, electronic mail). Broadband services require high bandwidth or bit-rates greater than 2 Mbits/second, e.g. video (video-telephony, broadcast TV, teleconference, pay TV, HDTV etc.) and long duration data (word-processing, home-computing, database access, computer communication etc.).

According to CCITT Rec.I.121, these services can be Continuous Bit stream Oriented (CBO) or cell oriented (bursty) depending on the way of generation of data on the source side. A CBO source generates cells periodically at regular intervals. On the other hand, a bursty source intermittently generates a series of cells. Within the B-ISDN, CBO services require much higher quality of service than do voice communications. The quality of CBO services has to be about the same as that for existing circuit switching networks, i.e. no delay fluctuations, cell loss rate of 10^{-9}, switching delay of 0.5 ms [CCITT G.142]. While it is difficult to guarantee no delay fluctuations in an ATM network, our scheme allows us to bound the delay and delay variation within the acceptable limits. Provision of performance guarantees to the two kinds of bearer services requires a scheme for resource allocation in an ATM framework.

Within the realm of bursty traffic itself, we can distinguish two types of traffic, delay-sensitive bursty traffic requires a switching delay of about 1 ms per switch, with the probability of a packet being delayed beyond this bound less than 10^{-9}; and loss-sensitive traffic which ideally should not lose any packets because of buffer overflows at any switch. Of course, perfect reliability can only be obtained by means of acknowledgments and retransmissions. Thus we can classify our performance requirements into the following three types:

Class 0 Traffic: is CBO, requires a delay of less than 0.5 ms per switching node and no packet losses due to buffer overflows.

Class 1 Traffic: is bursty, requires a delay of less than 1.0 ms per switching node and packet losses due to a missed delay bound or buffer overflows is less than 10^{-9}.

Class 2 Traffic: is bursty, requires no delay bounds but must have no packet losses due to buffer overflows.

Thus our problem is to devise a way to support this menu of QOS (Quality of Service) in the model of the ATM switch described next.

2.1. ATM switch

The ATM switch can be modeled as a set of input queues (incoming links) which are switched through an interconnection network to a set of output queues(outgoing links). Usually the only purpose of the input queues is cell/bit synchronization and clock recovery, so the input queue size is minimal. Nevertheless, it should be noted that statistical switching techniques will necessitate large input queue sizes in some cases. The output queues are needed since cells from different incoming links that are to be routed towards the same outgoing link may arrive concurrently.

A description of the different switch architectures and justification of the above model has been provided by Karol[Karo87]. For ease of presentation, we will assume that the ATM switch is internally non-blocking, cells arriving on different input links and destined for different output links do not interfere with each other, input queues are non-existent, and framing details can be ignored. It allows us to concentrate on one output link and devise a scheme to bound the queueing delay in the output port.

An ATM network consists of a number of switches connected in a random mesh. Each connection can be modeled as passing through a number of queueing servers, each server modeling the output link at an intermediate switch. We also assume that each switch is accompanied by a switch manager, which is responsible for accepting or rejecting connections through the switch. The switch manager should be running the algorithms and tests described later. When a connection is established, the switching fabric maps cells arriving along that connection from the input link to the appropriate output link. We are not concerned with the exact details of this mapping mechanism.

In the next two sections, we describe the scheme to provide quality of service in an ATM network that consists of a mesh of the switches described above. As mentioned in Section 1, this would involve describing the four areas of specification, rate-control, scheduling and admission control.

3. Specification:

The provision of performance guarantees by means of resource allocation can only be done efficiently if the clients specify their traffic characteristics and obey them when transmitting data. We impose the following restrictions on any such specification:

(1)The specification must be exact, i.e., the network must be able to verify whether the client is obeying his promises or not. Thus, specifications of traffic by means of probability density functions [Wang90] or complex correlation coefficients[Guse90] are ruled out since there is no known simple way to verify if the traffic is obeying the specification or not.

(2) Flexible: The specification must allow the clients to specify traffic with widely differing characteristics of burstiness, and bandwidth requirements.

(3) Simplicity: The specification must be simple and concise. The advantages of a simple specification are obvious.

(4) Maintainability: The traffic characteristics of any connection will change as we move along the path of a connection. In general, the burstiness of any connection tends to increase as we proceed in this fashion. Thus resources allocated to a connection in this fashion will tend to increase along the path. The network must be able to account for this change in traffic specification and ideally be able to preserve the same traffic specification from node to node.

Three simple traffic specifications have been proposed in existing literature for admission control. We consider them one by one.

(1) *The average rate model*: [Gole90] specifies the average bandwidth required by the connection over an averaging interval. This could be looked upon as an window-based mechanism, where at most r packets can be send in an averaging interval of length I. Alternately, we can look upon it as a rate-based specification, when the average spacing between any two packets is at least x_{ave} in every interval of length I (with the obvious equivalence that $x_{ave}=I/r$). A slight variation of the scheme would specify that the spacing between any two packet be at least x_{ave}.

The disadvantage of the average rate model is that it does not allow for burstiness in a client's traffic. Thus, this model must be augmented by models which allow for variable rates or bursts.

(2) *The average-rate burst-size model*: [Cruz87] specifies a maximum burst-size in addition to an average rate. In every time interval of length I, the number of packets on the connection must not exceed $I/x_{ave}+b_{\max}$, where x_{ave} is the average spacing between the packets and $b_{\max}$ is the maximum burst size. An intuitive explanation of the model is that it allows up to $b_{\max}$ packets to be transmitted back-to-back, but maintains a long-term average rate of I. [2]

This model allows some burstiness, reflected by the burst-size parameter. However, the burstiness (the burst size $b_{\max}$) increases as one moves across the nodes in a network. In terms of resource allocation, (using the resource allocation schemes of [Andr89]) this model tends to tax the nodes downstream much more heavily than the nodes upstream. We do not know of a simple way to maintain the burst size parameter across the network.

(3) *The peak-rate average-rate model*: [Ferr90] This model allows the sender to transmit at two different rates, a peak and an average rate. The minimum spacing between any two packets

[2] The actual specification by Cruz uses different terms than the ones we have described, but they are functionally equivalent. We have attempted to use the same terms in the specification of all the three specification model.

must be larger than $x_{\min}$ at all times (this corresponds to the peak rate); while the average packet spacing in any interval of length I must be x_{ave}. In this model, burstiness may be expressed as the ratio of peak to average bandwidth or $x_{ave}/x_{\min}$ [Choi89].

This model allows burstiness to be modeled, and has the advantage that the minimum and average spacings can be maintained at all nodes along the connection's path by a simple rate-control mechanism. This has obvious benefits for resource allocation schemes at the nodes downstream.

In general, the traffic generated by the clients in a B-ISDN network will not be well-behaved and the client will need a leaky-bucket mechanism at the entry of the network to convert the natural unregulated traffic into the regulated specified traffic adhering to the above mentioned models. The efficiency of such leaky bucket schemes is discussed in Section 5. from those simulations, it seems that the peak-rate average-rate model is best suited for specifying the traffic characteristic of connections in an ATM network at reasonable burstiness values, typically requiring burstiness values less than 5, and using an I about twenty times x_{ave}.

Along with the traffic specifications, the client desiring to communicate must also declare his performance requirements. In our model, he does so by declaring which of the three classes of service in Section 1 he belongs to.

For ease of presentation, we assume that the averaging interval I is a network-wide constant. It is possible to extend our resource allocation schemes to allow different averaging intervals for different connections, but the advantages of such extensions are not significant.

3.1. Rate control

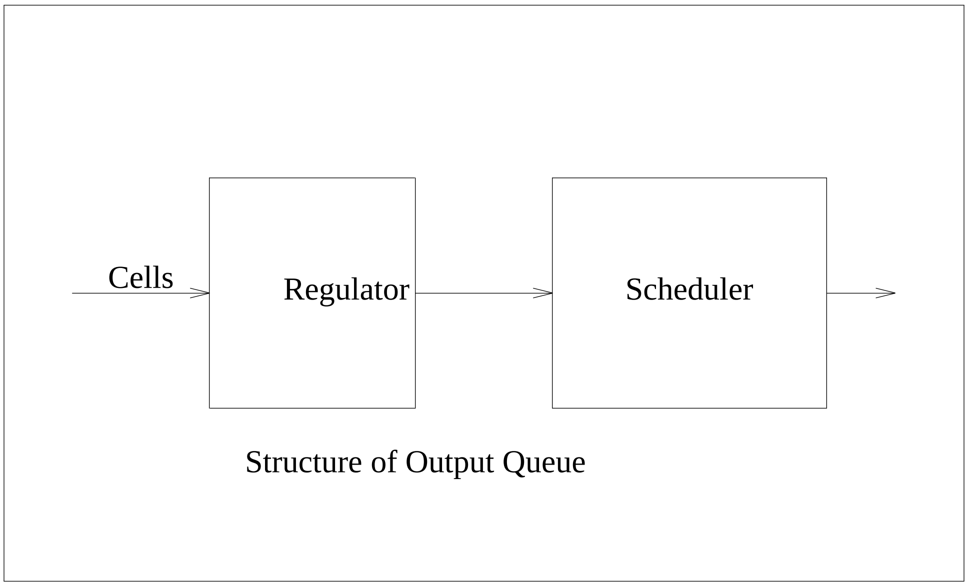
Enforcement of traffic characteristics is a must since a malicious user could send cells into the network at a much higher rate than the declared maximum or average value. The same effect might be caused by a failure in the sending host or in a switch itself. Network fluctuations can also cause the minimum inter-packet spacing along a connection to be less than the declared bound. If we do not take appropriate countermeasures, such malicious or faulty behavior can prevent the satisfaction of the delay bounds guaranteed to other clients of the real-time service, thereby damaging the clients and destroying the credibility of the service.

The enforcement of the traffic specification can be termed rate-control. Rate-control can be done by forcing the "offending" cells to wait till the time they were actually supposed to arrive. When buffer space is limited, packets that arrive too much in advance of their proper time might even be dropped because of buffer overflow. Of course, a sufficient amount of buffer space will have to be allocated to prevent the offending cells from flooding the buffer space of a heavily loaded switch and causing cells from other connections to be dropped.

Rate-control needs to be done at each output queue of an ATM switch. Thus, the output queue consists of two modules, a regulator which acts as a policeman preventing misuse of the network, and a scheduler which is responsible for selecting the cells to be transferred next on the output link. (See Figure 1).

Thus, rate control requires that we calculate the expected arrival time of each packet along a connection, which is the time the packet should have arrived if it was obeying its $x_{\min}$ and x_{ave} constraints. Calculation of this expected arrival time is simple and can be done by means of the following simple pseudo-code (adopted from [Ferr89]).

Figure 1




```
initialize()
{
        count = 0;
        expected_time = clock;
        next_I = clock + I;
}
```

On the acceptance of a connection, three kinds of information (the state) needs to be kept. The state information predicts the earliest time the next packet on a connection can arrive. The information can be initialized by means of a routine like `initialize()`.

On the arrival of every packet the procedure `packet_arrival()` is called, it gives the time the packet should actually have arrived at the node, obeying all the traffic specifications. If the packet happens to have arrived before it is expected, it is held for the period of time equaling the difference between expected arrival time and the current time, before being handed over to the scheduler.

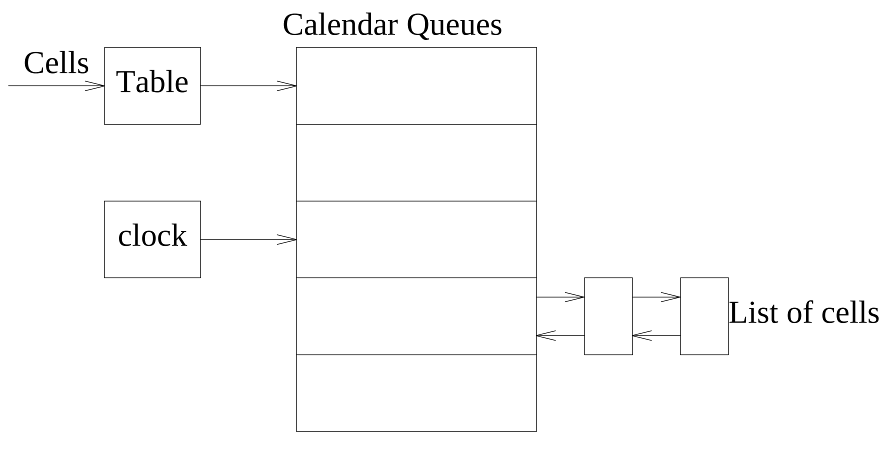
The hardware implementation of the rate-control scheme may be done by means of a number of calendar queues[Brown88]. For the time-being, assume that we have an infinite number of such calendar queues. Calendar queue i would contain a list of the packets that were expected at time i, but may have arrived early. When the clock reaches the values i, all these packets are transferred to the scheduler by appending this list to the list of packets in the scheduler queue. The operation can be done in a constant number of steps. Since, the computation of the expected arrival times is done so as to preserve the $x_{\min}$ and x_{ave} constraints, the two properties are enforced at every switch along a connection's path.

In a practical implementation, we can only have a finite number of calendar queues. The effect of infinite calendar queues can be mimiced closely if the number of such queues is large enough to keep all the packets that would arrive in the current interval of length I. Suppose there

```
packet_arrival()
{
        count++;
        if (clock > expected_time) expected_time = clock;
        /* The value of expected time now is the one for this packet */
        answer = expected_time;

        /* Compute the expected time for the next packet */
        if (count >= I/x_ave) {
                expected_time = max(next_I, answer + x_min);
                count = 0;
                next_I += expected_time + I;
        } else {
                expected_time += x_min;
        }
        return(answer);
}
```

Figure 2



Structure of Regulator

are N calendar queues. Also assume that the time axis is discrete and divided into slots of length t, the time required to transmit one ATM cell on the output link. Then, we need to ensure that $Nt \geq I$. Packets that arrive too far in the future (more than N time cycles ahead in the given example) would be dropped. As the clock advances, calendar queues which become empty can be used for holding the packets expected later in a wrap-around fashion.

The schematic diagram of rate-control can be seen in Figure 2. Note that the rate-control algorithm has to be performed at each switch to ensure that the traffic specifications are always satisfied. Also note that all the operations involved in rate-control can be done in a constant number os steps. Since there is no list processing or priority queue manipulation, the rate-control scheme can be done very easily in hardware. [3]

4. Scheduling and Admission Control

Keeping in mind that the delay requirements of the CBO connections are more stringent than the delay requirements of bursty traffic, the first choice of a scheduling scheme appears to be one that gives priority to the CBO traffic over the bursty ones. Such a scheme is proposed by Murase [Mura89] where the authors show that it minimizes the influence of the bursty traffic on the CBO traffic. Since bursty traffic itself consists of two categuories, the delay-sensitive bursty traffic must be given preference over other kinds of bursty traffic.

In each ATM switch, three queues are maintained at each output port: one for cells belonging to each class. When the node must choose the next cell to be transmitted, the scheduler picks a class 0 cell, if one is present. Otherwise any existing class 1 cell is transmitted. Class 2 cells are transmitted only when no cells from any of the upper classes are present. This scheduling mechanism results in good delay characteristics for delay-sensitive traffic[Todo90].

This is a simple priority scheduling mechanism, which can be done at the high speeds required for B-ISDN networks. Certainly, this is not the optimum scheduling scheme for an ATM network, complex algorithms will in general perform much better [Fitz89][Chen89]. However, these algorithms may not be feasible at higher speeds.

4.1. The admission control tests

When a switch receives an establishment request message, it performs some or all of the following tests:

the class 0 bandwidth test, required when the connection to be established belongs to class 0, and involving the existing class 0 connections sharing the same output link;

the class 1 bandwidth test, to be performed for the establishment of both class 0 and class 1 connections if the new connection belongs to class 1, or if at least one class 1 connection shares the same output link; this test involves all class 0 and class 1 connections on the output link;

the class 2 bandwidth test, to be performed in all conditions;

the delay bound tests for class 0 and 1 verify if the delay constraints of the class 0 and class 1 traffic are being met.

the buffer space test, which is needed for all types of connections in order to determine if sufficient buffer space in present in the switch to accommodate the new connection.

If the request passes the tests, then the switch manager sends the establishment message on to the next switch. Otherwise the request is rejected.

The class 0 bandwidth test consists of verifying that the outgoing link is fast enough to accommodate the additional class 0 connection without impairing the guarantees given to the others. Keeping in mind that class 0 connections have a constant bandwidth requirement ($x_{\min} = x_{ave}$), we simply need to check that

$$\sum_j t/x_{min,j} < 1, \tag{1}$$

where the sum is taken over all the class 0 connections sharing the output link with the incoming

[3] The rate-control process has some similarities with a rate-based scheduling mechanism referenced in [Kana90].

request. We would like to remind the reader that t is the time to transmit one cell on the output link.

The class 1 bandwidth test tries to statistically multiplex a number of bursty connections to achieve better switch utilization. It is responsible for determining whether the probabilistic delay guarantees for bursty connections already accepted by the switch are safe (within the bound of 10^{-9}) that is required by the QOS guarantees.

Let us divide the time-axis into regions where the bandwidth demanded by all the active connections is less than the bandwidth of the outgoing link, and into regions where it is not. When the bandwidth requested is more than the link's bandwidth, we are operating in an unsafe region. Define the overflow probability as the probability that the we are likely to enter such a situation. An upper bound on the probability that a class 1 connection will be delayed beyond its delay bound is given by this overflow probability.

The probability that connection j is *active* (i.e., is carrying packets) at some instant of time during an interval I is

$$p_j = x_{min,j} / x_{ave,j}. \tag{2}$$

Given K independent connections passing through a node, the probability that the members of a subset C of them are simultaneously active is given by

$$Prob(C) = \prod_{i \in C} p_i \prod_{j \notin C} (1 - p_j). \tag{3}$$

Let at least one of the connections be a class 1 connection. To compute P_o, the overflow probability at the node, we start by listing all the *overflow combinations*. An overflow combination is a set of connections that, when simultaneously active requires a bandwidth larger than that of the outgoing link. In other words, an overflow combination is one for which inequality (1) above is not satisfied.

The bursty test consists of computing this probability and verifying that the value does not exceed the threshold required by class 1 service. In other words, the overflow probability should remain less than the value of 10^{-9} required for class 1 grade of service. Although a brute-force evaluation of this probability will take exponential time, a good approximation can be obtained in a constant amount of time (if the accuracy required is fixed). We shall not mention the approximation algorithm in order to keep the paper short.

The delay bound tests verify that the delay guarantees of the class 0 and class 1 connections can be met whenever the switch is not overloaded. If there are N_0 connections which belong to class 0 grade of service, then equation (1) implies that the maximum delay of a class 0 cell can not exceed d_0 [Cid88] where

$$d_0 = (N_{cbo} + 1)t, \tag{4}$$

where t is the basic cell transmission time on the output link.

Let us consider the largest combination of class 0 and class 1 connections that satisfies condition (1). This set will consist of all the N_0 class 0 connections on the output link and some N_1 low delay burst mode connections, selected in the decreasing order of $x_{\min}$ values. The maximum delay of any class 1 cell, when operating in the safe zone of the time axis is bounded above by d_1, given by

$$d_1 = \left\lceil \frac{N_0 + N_1}{(1 - \rho_0)} \right\rceil t + t, \tag{5}$$

where ρ_0 is the utilization of the output pipe by all the existing class 0 connections. In other words, $\rho_0 = \sum t/x_{ave,i}$, with i ranging over all the class 0 connections.

The delay bound tests consists of verifying that both d_0 and d_1 are within the acceptable limits of their respective grades of service. Thus d_0 must be less than 0.5 ms and d_1 must be less than 1.0 ms.

Buffer allocation is required in order to guarantee connections with a bounded loss rate. It is also known that a shared buffer architecture results in a much better buffer utilization than an architecture in which each connection is statically allocated a fixed number of buffers. Notice that the rate-control scheme prevents greedy clients from usurping the whole of the shared buffer space. Thus it is safe to have connections share a common buffer space.

What is the total buffer requirement of all the connections sharing the same communication links. Let us look at the buffer requirements of the scheduler first. Let us examine the state of the scheduler at a time T after it makes a transition from an idle to a busy state, then the count of cells present in the scheduler queue can be bounded above by the amount C_T given by

$$C_T = \sum_i \min(\lceil T/x_{min,i} \rceil, \lceil I/x_{ave} \rceil) - T, \tag{6}$$

where i ranges over all the connections sharing the output link (and including) the connection being established. It can also be shown that the maximum value of C_T will occur either at $T=0$ or at $T = \lceil I/x_{ave,i} \rceil t$ where i could be any of the connections sharing the output link. Thus, with N connections sharing an output link, the maximum buffer requirement in the number of cells can be computed by obtaining the maximum value of equation (6) evaluated at the the $N+1$ values of T.

Keeping in mind the fact that the maximum number of cells a connection can have in the regulator is the maximum number of cells that are present in the scheduler immediately before the node, it follows that in a network of homogenous switches, the buffer requirement of the regulator is the same as the buffer requirements of the scheduler. Thus, the buffer space test consists of evaluating the maximum value of equation (6) at the $N+1$ values of T mentioned in the previous paragraph, doubling this value to account for the regulator, and then verifying that the buffer space in the switch exceeds this computed amount. Of course, this test assumes that the regulator and the scheduler share the same buffer space as well.

These admission control criterion are simple and easy to enforce at the connection establishment time with limited amount of processing power. However, the astute reader would have noted that both the delay and buffer computations are done on the basis of a worst-case assumption, assuming that all connections may be active at the same time. There is a cost to worst-case design in terms of poor network utilization, however it is the only safe way if we are to ensure that we meet the required quality of service.

5. The Simlations

In this section, we present the results of the simulation experiments we ran to study the performance of our admission control algorithms. We were interested in a number of issues:

(1) Is the scheme correct? Can we discover cases in which the admission control schemes would fail to produce the desired results?

(2) Is the specification model used satisfactory? Can reasonable values of I be obtained? can unregulated traffic be converted easily to the regulate traffic? What are the typical burstiness