Report Number: WUCS-93-31

1993-01-01

Real-time Admission Control Algorithms with Delay and Loss Guarantees in ATM Networks

Apostolos Dailianas and Andreas D. Bovopoulos

A multimedia ATM network is shared by media streams with different performance requirements. For media streams such as file transfers, the preservation of bursts and the provision of guarantees for loss probability at the burst level is of primary importance, while, for media streams such as voice, loss guarantees at the cell level are sufficient. Continuous media have stringent delay jitter requirements. Finally, some applications require loss-free transmission. In this paper, the first complete traffic management scheme for multimedia ATM networks is introduced. The traffic management scheme supports four different classes of traffic, each of which has different performance... **Read complete abstract on page 2.**

Real-time Admission Control Algorithms with Delay and Loss Guarantees in ATM Networks

Apostolos Dailianas and Andreas D. Bovopoulos

Complete Abstract:

A multimedia ATM network is shared by media streams with different performance requirements. For media streams such as file transfers, the preservation of bursts and the provision of guarantees for loss probability at the burst level is of primary importance, while, for media streams such as voice, loss guarantees at the cell level are sufficient. Continuous media have stringent delay jitter requirements. Finally, some applications require loss-free transmission. In this paper, the first complete traffic management scheme for multimedia ATM networks is introduced. The traffic management scheme supports four different classes of traffic, each of which has different performance requirements expressed in terms of delay jitter guarantees and cell or burst level loss guarantees. Its running time specification as well as its real time admission control algorithms are completely specified.

Real-time Admission Control Algorithms with Delay and Loss Guarantees in ATM Networks

Apostolos Dailianas and Andreas D. Bovopoulos

WUCS-93-31

August 1993

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130-4899

Real-time Admission Control Algorithms with Delay and Loss Guarantees in ATM Networks

Apostolos Dailianas and Andreas D. Bovopoulos
Computer and Communication Research Center
Washington University in St. Louis

Abstract

A multimedia ATM network is shared by media streams with different performance requirements. For media streams such as file transfers, the preservation of bursts and the provision of guarantees for loss probability at the burst level is of primary importance, while, for media streams such as voice, loss guarantees at the cell level are sufficient. Continuous media have stringent delay jitter requirements. Finally, some applications require loss-free transmission. In this paper the first complete traffic management scheme for multimedia ATM networks is introduced. The traffic management scheme supports four different classes of traffic, each of which has different performance requirements expressed in terms of delay jitter guarantees and cell or burst level loss guarantees. Its running time specification as well as its real time admission control algorithms are completely specified.

1. Introduction - Motivation

Asynchronous Transfer Mode (ATM) is the technology of choice to drive the enterprise network into the twenty-first century. ATM can seemlessly integrate all components of an enterprise network-LANs, MANs, and WANs-and can serve as a transparent backbone technology for other technologies like Frame Relay and SMDS. ATM backplane-based internetworking modules allow high speed bridging and routing between Ethernet, Token Ring and FDDI segments and connectivity across the WAN. Being a switched and not a shared medium based technology, ATM supports the creation of virtual LANs so that users can be associated into groups that are independent of the physical network topology, an approach more suited to work-group oriented enterprise organization and client-server architectures. The availability of such a flexible networking infrastructure will accelerate the deployment of value-added multimedia applications. However, an ATM infrastructure cannot support efficiently multimedia applications without the provision of traffic engineering rules capable of providing performance guarantees to the diverse media types that will be using it.

Some virtual circuits (VCs) transmit correlated sequences of cells, called bursts, generated because of the physical (source dependent) or logical (protocol dependent) structure of the information stream carried by the VC. Losing a number of cells of a burst may result in the loss of the information content of the whole burst. Therefore, an ATM network should support VCs requesting QoS guarantees with respect to the burst-loss probability. This problem has been studied in [2, 15, 11].

An ATM network should also support efficiently VCs requiring performance guarantees at the cell level. The estimation of cell-loss probability for networks using statistical multiplexing is one of the most studied problems in the ATM literature [10, 8, 13, 6, 5, 1]. In order to calculate the cell-loss probability, it is necessary to calculate the steady-state occupancy of the buffer, obtain the probability of buffer overflow, and relate it to the loss probability of each VC. This approach is computationally very expensive, not allowing real time decisions to be taken; thus, it is only of theoretical interest. To overcome the computational complexity of the previous solutions, a number of proposed solutions are based on a bufferless model [6, 12] and thus provide an upper bound with respect to the expected cell loss probability.

Finally, in terms of provision of loss guarantees, an ATM network should support VCs demanding a loss-free connection. The solution proposed in the literature is peak allocation for the whole duration of the connection. When the VCs requesting loss-free transmission are bursty in nature, peak allocation results in underutilization of the network resources. Admission control algorithms that exploit the statistical characteristics of the VCs requesting loss-free transmission would result in better utilization of the network resources.

In a multi-media application, continuous media (CM) streams, such as real time video, have stringent delay jitter requirements. The term *jitter* refers to the distortion introduced to the original traffic pattern by the consecutive queueing stages along the path of the connection. It is quantitatively measured as the maximum difference between the end-to-end delays of two consecutive cells of the same stream. For such VCs, an ATM network must provide jitter guarantees, independent of the number of nodes the media stream traverses from source to destination.

From the above discussion it is clear that an ATM network should be able to support VCs requiring performance guarantees in terms of loss at the cell and/or burst level and, if necessary, guarantees in terms of the maximum delay jitter that the ATM network can introduce to the cells of a VC. All the above mentioned options for QoS should be incorporated into a real-time connection admission control (CAC) algorithm.

CAC algorithms use the traffic characteristics declared by a source in the connection admission process and usually assume that the traffic characteristics of the source do not change along the path of the VC. The source traffic characteristics do change, however, as a result of the buffering performed at each switch. The change can result in the total alteration of the initial structure of the stream [3, 7, 17] which in turn might invalidate the results of the CAC algorithm for downstream nodes. Proposed solutions for the prevention of the alteration of the traffic characteristics of a VC are based either on the reconstruction of the initial structure at every node through dedicated hardware mechanisms [3] or on the restriction of the change of traffic characteristics through the use of framing [7].

This paper proposes a new traffic management scheme (TMS) that supports four different types of service with various types of QoS guarantees (Table 1). To our knowledge

Guarantees	Loss-free Service (Type A)	Burst-scale Service (Type B)	Cell-scale Service (Type C)	Unreliable Service (Type D)
Delay and jitter	$\surd$	$\surd$	$\surd$	-
No loss	$\surd$	-	-	-
Burst loss probability	-	$\surd$	-	-
Cell-loss probability	-	-	$\surd$	-
Cell-loss probability (soft)	-	-	-	$\surd$

Table 1: Classification of Provided Services

this is the first network management scheme proposal to support such a variety of service types and QoS guarantees through a real time CAC algorithm with a reasonable hardware complexity.

The rest of the paper is structured as follows. In Section 2 the framing strategy which is essential for the provision of jitter bounds is introduced. Section 3 discusses the characterization of the traffic streams and provides an extension to the framing strategy introduced in the previous section. In Section 4 the proposed TMS is introduced, and Section 5 describes its running time operation for Type A, B and C VCs. This is an event driven specification of the actions that the system takes upon the arrival of a cell and is primarily distinguished on the basis of the service provided to the stream to which the cell belongs to. Section 6 presents the admission control algorithms for the acceptance or rejection of an incoming call of Type A, B or C. Section 8 provides numerical examples and a discussion of the main parameters affecting the operation of the TMS. Section 9 deals with mechanisms for the policing of the traffic parameters specified by the user at connection setup time. Section 10 presents the running time specification and the connection admission algorithms for Type D VCs.

2. Delay Jitter Control with Time Framing

As explained in the previous section, a CAC algorithm uses the traffic characteristics of a source in making the decision as to whether a particular switching system along a VC can support a connection. Framing of time is a mechanism that can provide (i) preservation of traffic characteristics of a VC and (ii) guarantees in terms of delay and delay jitter [7].

In the framing strategy proposed in [7], the traffic of a particular VC enters a switch at a particular port *in* and exits the switch from a particular port *out*. Let R_l be the link rate at the output port. A framing strategy and a particular link stream is divided into consecutive and equal frames of duration T, as depicted in Figure 1. Note that the frames of the input port do not have to be aligned in time with the frames at the output port. Their misalignment is captured by the parameter Θ ($\Theta \epsilon [0, T)$), which corresponds to the time difference between the end of a frame of the input port and the beginning of the next frame of the output port.

In the framing strategy, a VC must specify a rate $\tilde{r}_H$, which corresponds to the maximum number of cells that the VC can transmit within each frame. An incoming cell departs during the first frame that starts after the end of the frame of the incoming cell. The hardware implementation of the framing strategy is simple and is based on two FIFO queues.

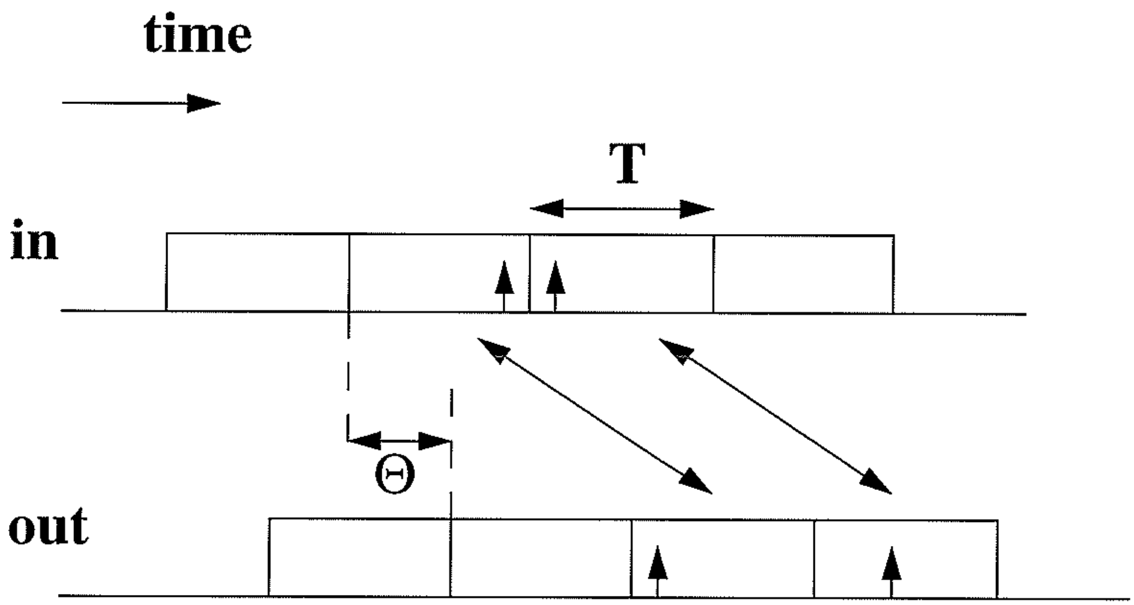


Figure 1: Framing Strategy

The strategy can guarantee that the end-to-end delay of a cell is a constant plus a small bounded jitter term. The end-to-end delay is given by $D = \hat{D} + d$, where $\hat{D}$ corresponds to the queueing, propagation, processing and switching delays of a connection and where d is the displacement of the cell, *i.e.* the difference between its position in the frame at the sender and its position in the frame at the receiver. From the definition of d, it holds that $-T < d < T$, which is independent of the number of nodes H through which the VC passes. The term $\hat{D}$ can be further decomposed into two terms as $\hat{D} = \hat{T} + Q$. $\hat{T}$ consists of the propagation, processing and switching delays and is independent of the frame size T, while Q is the sum of the queueing delays and lies in the interval $[H \cdot T, 2 \cdot H \cdot T]$. Note that Q is constant for each connection. Referring to Figure 1, the jitter is defined as the maximum difference between the end-to-end delay of any two cells.

The choice of the frame size T is subject to three constraints. First, if R_l is the link rate at the output port of the switch, then $3 \cdot R_l \cdot T$ should be less than or equal to the available buffer size. This condition is explained by the fact that at any instant, the cells of at most 3 frames (the incoming frame, the outgoing frame and the frame corresponding to the time difference Θ) are stored in the buffer. Second, the jitter is between $-T$ and T. Finally, there is a direct relation (shown by the queueing delay Q) between the value of T and the maximum end-to-end delay that the media stream can tolerate.

In order to get a better feeling for the relation between the frame size, the buffer requirements, the corresponding maximum queueing delay per node and the provided jitter guarantees, refer to Table 2, where link rate of $155 Mbps$ is assumed.

T in μsec	T in cells	Required buffer in cells	max. queueing delay per node in μ sec	jitter in μ sec	jitter in cells
90.3	33	99	180.6	(-90.3,90.3)	(-33,33)
273.5	100	300	547	(-273.5,273.5)	(-100,100)
454	166	498	908	(-454,454)	(-166,166)

Table 2: Relation between frame size, buffer, delay and jitter

There are two potential drawbacks that must be addressed if the framing strategy proposed in [7] is to be used in the context of a CAC algorithm. First, the traffic characteristics and QoS requirements of a particular VC must be related to its framing strategy. Second, the framing introduces a discretization of possible rates of operation of a source. This second drawback is explained further in Section 3. The first of these problems is the dominant source of inefficiency, but the second can also become important, especially for small frame sizes. In this paper a CAC algorithm is introduced, based on the framing strategy proposed in [7], but enhanced through the successful solution of both of the above mentioned problems.

3. Traffic Characterization in the Presence of a Framing Mechanism

The traffic characteristics of the traffic stream i are specified by two parameters, the *peak rate* λ_i and the *mean rate* μ_i, both in *bps*. These characteristics are chosen since they are general enough to apply to a very broad range of possible traffic streams making use of the network. To use the framing strategy described, the peak and mean rate of the source must be mapped to the rate r_{H_i} for a given frame size T. Let T_i be the minimum distance between consecutive cells of source i ($T_i = 48 \cdot 8/\lambda_i$ sec). Then the maximum number of cells of source i within T is $\lceil T/T_i \rceil$, resulting in:

$$r_{H_i} = \lceil \frac{T}{T_i} \rceil \cdot r_{step} \quad , \tag{1}$$

where r_{step} corresponds to the rate of a source transmitting one cell per frame and is given by

$$r_{step} = 53 \cdot 8/T \ bps \quad . \tag{2}$$

If $\lceil T/T_i \rceil \neq T/T_i$, this approach overestimates the maximum rate of source i by $(\lceil T/T_i \rceil - T/T_i) \cdot r_{step}$. In order to overcome the potential maximum-rate overestimation, two rates are assigned to the source when active, $r_{H_i} = \lceil T/T_i \rceil \cdot r_{step}$ and $r_{L_i} \stackrel{\text{def}}{=} r_{H_i} - r_{step}$, with probabilities p_{H_i} and p_{L_i} respectively. In general $p_{L_i} = 1 - p_{H_i}$, and

$$p_{H_i} = \begin{cases} T/T_i - \lfloor T/T_i \rfloor & \text{if } \lceil T/T_i \rceil \neq T/T_i \\ 1 & \text{otherwise} \end{cases}$$

The two-rate approach eliminates the inefficiency introduced by the discrete steps of bandwidth allocation due to the frame size T. On the the other hand, the two-rate approach introduces some uncertainty as to whether the number of cells of stream i arriving during a frame period T will be $\lceil T/T_i \rceil$ or $\lfloor T/T_i \rfloor$. As will be clarified later, because of the reservation performed during the operation of the system, in order to provide guarantees for the loss-free and burst-scale sources (Type A and B of service), both of these services cannot tolerate the uncertainty introduced by the two-rate mapping and use the single-rate approach, while the cell-scale service (Type C) will use the two-rate approach.

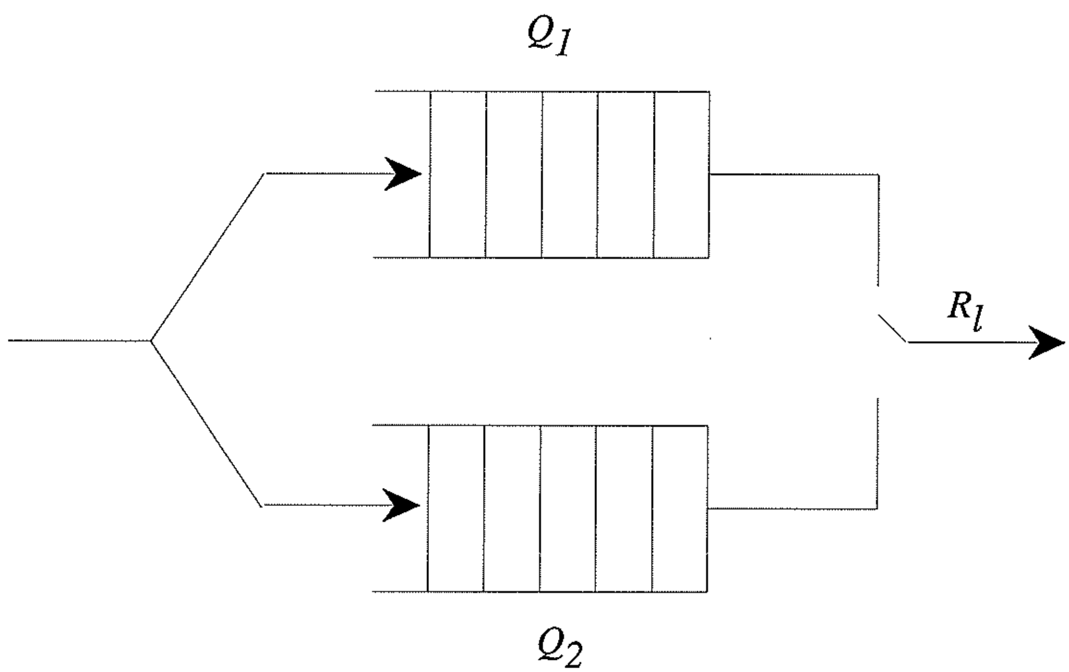


Figure 2: Proposed traffic management scheme

4. Traffic Management Scheme

The traffic management scheme proposed in this paper supports the four types of service outlined in Table 1. Each switching system is logically organized in the form of two queues, as depicted in Figure 2. Q_1 is intended for applications of Types A, B and C. All cells passing through Q_1 are guaranteed a bounded delay and delay jitter. Q_2 is intended for applications of Type D, which do not have stringent loss, delay or delay jitter requirements. Cells from Q_2 are serviced whenever there are empty slots in the output frame which are not used by eligible cells from Q_1. Filling only these empty slots guarantees that VCs of Type D do not interfere with the traffic of VCs of Types A, B, and C. Finally, R_l signifies the available link rate. Due to the discretization of bandwidth allocation in multiples of r_{step}, the users observe a link rate $\hat{R}_l$ which is equal to

$$\hat{R}_l \stackrel{\text{def}}{=} \lfloor \frac{R_l}{r_{step}} \rfloor \cdot r_{step} \quad .$$

To simplify the notation, let

$$\check{R}_l \stackrel{\text{def}}{=} \frac{\hat{R}_l}{r_{step}},$$

$$\check{T}_i \stackrel{\text{def}}{=} \frac{T}{T_i} \quad ,$$

$$\check{r}_{L_i} \stackrel{\text{def}}{=} \lfloor \check{T}_i \rfloor,$$

and

$$\check{r}_{H_i} \stackrel{\text{def}}{=} \lceil \check{T}_i \rceil \quad .$$

5. Running Time Specification of Q_1

The running time specification defines the actions to be performed upon the arrival of a cell at a switch. The running time specification described in this paper requires knowledge of the state of all sources in order to specify the instantaneous cumulative rate of the sources. The burst identification method proposed in [15] is used for this purpose. Every cell is marked as the *start, middle* or *end* cell of the burst. The arrival of a *start* cell at a switch changes the state of the VC from idle to active, the arrival of a *middle* cell maintains the state of the VC at active, and the arrival of an *end* cell changes the state of the VC to idle. A transition from active to idle is also made if no cell is received within a fixed timeout period.

In order to define the actions to be taken, let $\check{r}_{rsvd}$ represent the number of slots per frame reserved at any moment for VCs of loss-free and burst-scale sources.

5.1. Type A: Loss-free Service

An arriving cell of a loss-free source must always be admitted. In order to guarantee no loss, reservation of bandwidth must be performed upon the arrival of a new burst (*start* cell). The reserved bandwidth is released at the end of the burst (*end* cell).

If, upon the arrival of a *start* cell from source i, there is not enough bandwidth available (that is, if $\check{r}_{rsvd} + \check{r}_{H_i} > \check{R}_l$), then bursts of Type B VCs are dropped until there is enough bandwidth for the accommodation of the burst of Type A VC. A burst can be dropped from a switch by changing one of its cells from *middle* or *start* to an *end* cell. The choice of the burst of Type B to be dropped is arbitrary, but the decision process can be optimized using information such as the number of nodes the burst has thus far passed through, the number of nodes remaining to be crossed up to the receiver, and the rate and size of the burst.

A truncated burst of a VC of Type B is considered as lost in the CAC algorithm specifying the burst loss probability for a VC of Type B.

5.2. Type B: Burst-scale Service

For the sources for which burst loss probability is a concern, the decision to accept to Q_1 is performed on a burst basis. Upon the arrival of a *start* cell from source i, the new burst is accepted if there is enough bandwidth available (that is, if $\check{r}_{rsvd} + \check{r}_{H_i} \leq \check{R}_l$); otherwise it is rejected. If the burst is accepted, all cells (*middle*) up to the end of the burst (*end* cell) are accepted; otherwise they are rejected.

5.3. Type C: Cell-scale Service

For the sources for which the cell loss probability is a concern, the decision to accept to Q_1 is performed on a cell-basis. If, at the beginning of an output frame, $\check{r}_{rsvd}$ slots are reserved for the service of Type A and B VCs, $\check{R}_l - \check{r}_{rsvd}$ slots are available for the service of Type C sources. The running time operation is then as follows. During each frame, a counter counts the total number of cells of Type C that have been accepted since the beginning of the frame. Every time a cell of Type C arrives, if the counter is less than $\check{R}_l - \check{r}_{rsvd}$, the cell is accepted; otherwise the arriving cell is rejected.

6. Connection Admission Control (CAC) Algorithm

6.1. CAC Algorithm for Type B (Burst-scale) Service

The algorithm proposed in [15] for the computation of the burst loss probability based on the availability of the required buffer slots is also used here. However, rather than focusing on the availability of buffer slots, in this paper the focus is on the availability of the bandwidth required to support the new burst.

Let x_i be the random variable representing the rate requirements (in slots/frame) of burst-scale-source i at a random instant, and let X be the random variable representing the total rate requirements, including the source requesting connection. X is given by $X = \sum_{i=1}^{n} x_i$, where n is the total number of sources accepted by a node plus the source under consideration. Also, let p_i be the probability that source i is active at a random instant and $\bar{p}_i$ be the probability that it is idle. Then $p_i = \mu_i/\lambda_i$, $\bar{p}_i = 1 - p_i$.

Let

$$C_j^b \stackrel{\text{def}}{=} Prob\{\text{Rate of active bursts-scale streams in slots/frame} = j\}$$

and

$$S_j \stackrel{\text{def}}{=} Prob\{\text{Rate of active bursts-scale streams in slots/frame} > j\}$$

Then,

$$S_j = S_{j+1} + C_j^b$$

for all $j \geq 0$. Every time a new source is connected or an existing source is disconnected, the coefficients C_j^b for all j must be updated to reflect this change. Given the original coefficients C_j^b and the traffic characteristics λ_i, μ_i of the source i causing the change in the state of the system, the new coefficients $\{\hat{C}_j^b\}$ when the source is connected can be obtained by:

$$\hat{C}_j^b := C_j^b \cdot \bar{p}_i + C_{j-\check{r}_{H_i}}^b \cdot p_i \tag{3}$$

When source i is disconnected, the new coefficients $\{\hat{C}_j^b\}$ can be obtained as follows:

$$\hat{C}_j^b := \frac{C_j^b - \hat{C}_{j-\check{r}_{H_i}}^b \cdot p_i}{\bar{p}_i} \tag{4}$$

where all coefficients with negative subscripts are considered to be zero in the above equations.

The probability $Pr(X - x_i > \check{R}_l - \check{r}_{H_i})$ is an upper bound on the burst loss probability of virtual circuit i and can be computed as follows:

$$Pr(X > \check{R}_l) \;=\; \bar{p}_i \cdot Pr(X - x_i > \check{R}_l) + p_i \cdot Pr(X - x_i > \check{R}_l - \check{r}_{H_i})$$

Therefore

$$Pr(X - x_i > \check{R}_l) \;=\; \frac{1}{\bar{p}_i} \cdot Pr(X > \check{R}_l) - \frac{p_i}{\bar{p}_i} \cdot Pr(X - x_i > \check{R}_l - \check{r}_{H_i}) \quad .$$

After K recursive substitutions of the above equation, the following equation is obtained:

$$\begin{aligned} Pr(X - x_i > \check{R}_l) &= \frac{1}{\bar{p}_i} \cdot \sum_{h=0}^{K-1} (-\frac{p_i}{\bar{p}_i})^h \cdot Pr(X > \check{R}_l - h \cdot \check{r}_{H_i}) + (-\frac{p_i}{\bar{p}_i})^K \cdot Pr(X - x_i > \check{R}_l - K \cdot \check{r}_{H_i}) \\ &= \frac{Z_K}{\bar{p}_i} + (-1)^K \cdot W_K \end{aligned}$$

for any $K \geq 0$, where

$$Z_K = \sum_{h=0}^{K-1} (-\frac{p_i}{\bar{p}_i})^h \cdot Pr(X > \check{R}_l - h \cdot \check{r}_{H_i})$$

and

$$W_K = (\frac{p_i}{\bar{p}_i})^K \cdot Pr(X - x_i > \check{R}_l - K \cdot \check{r}_{H_i})$$

If $K = 1 + \lfloor \check{R}_l / \check{r}_{H_i} \rfloor$,then:

$$Pr(X - x_i > \check{R}_l) = \frac{Z_{1+\lfloor \check{R}_l/\check{r}_{H_i} \rfloor}}{\bar{p}_i} + (-1)^{1+\lfloor \check{R}_l/\check{r}_{H_i} \rfloor} \cdot (\frac{p_i}{\bar{p}_i})^{1+\lfloor \check{R}_l/\check{r}_{H_i} \rfloor}$$

Therefore,

$$Pr(X - x_i > \check{R}_l - \check{r}_{H_i}) = \frac{1}{\bar{p}_i} \cdot Z_{\lfloor \check{R}_l/\check{r}_{H_i} \rfloor} + (-\frac{p_i}{\bar{p}_i})^{\lfloor \check{R}_l/\check{r}_{H_i} \rfloor} \quad (5)$$

where

$$Z_{\lfloor \check{R}_l/\check{r}_{H_i} \rfloor} = \sum_{h=0}^{\lfloor \check{R}_l/\check{r}_{H_i} \rfloor - 1} (-\frac{p_i}{\bar{p}_i})^h \cdot S_{(\check{R}_l - (h+1)\cdot \check{r}_i)}$$

Computational Complexity: Consider the time that is required to check if a new request can be accepted. In order to update the set of coefficients $\{C_j^b\}$ when a new connection is requested, two multiplications and one addition must be performed for each coefficient. Given that the burst loss probabilities must be kept low in order to accept a new connection, the coefficients must decrease rapidly as j increases. Neglect therefore all coefficients with $j > 4 \cdot \check{R}_l$. Assume also that the set of values $\{S_j\}$ is computed and stored on-line. For each $0 \leq j \leq 4 \cdot \check{R}_l$, the computation of the value S_j requires one addition since $S_j = S_{j+1} + C_j^b$. Assume also that, in the computation of Z_K, the values $(-\frac{p_i}{\bar{p}_i})^h$ are computed recursively and stored on-line for any h. Then for each value of h it is necessary to perform two multiplications $((-\frac{p_i}{\bar{p}_i})^{h-1} \cdot (-\frac{p_i}{\bar{p}_i})$ and $(-\frac{p_i}{\bar{p}_i})^h \cdot (S_{\check{R}_l-(h+1)\cdot\check{r}_i})$. Thus, in order to compute the burst loss probability for source i, approximately $2 \cdot \check{R}_l / \check{r}_{H_i}$ multiplications must be performed. If n is the number of sources, including the source requesting connection, the admission decision must be performed for all n sources, since the new source must not violate the performance requirements of the already accepted sources. Therefore, the complexity of the decision process is approximately

$$2 \cdot 4 \cdot \check{R}_l + \sum_{i=1}^{n} 2 \cdot \frac{\check{R}_l}{\check{r}_{H_i}} \text{ multiplications and } 4 \cdot \check{R}_l + \sum_{i=1}^{n} \frac{\check{R}_l}{\check{r}_{H_i}} \text{ additions}$$

For $n = 200$ sources, $\lambda_i = 1\ Mbps$, $R_l = 155\ Mbps$ and $T = 0.456msec$ corresponding to 166 cells per frame and a buffer size of $3 * 166 \approx 500cells$, $r_{step} = .93Mbps$, $\check{r}_{H_i} = 1.86Mbps$, $\check{R}_l = 154.38Mbps$, approximately 34.6K multiplications and 17.5K additions must be performed. In a 30 MFLOPS processor (peak rate), this would require roughly $1.8msec$, or about $9\mu sec$ per source. In the case $T = 3\ msec$, which corresponds to 1096 cells per frame and a buffer size of almost 3,300 cells, $r_{step} = 141.3Kbps$, $\check{r}_{H_i} = 1.13Mbps$ $\check{R}_l = 154.9Mbps$ and $\check{R}_l = 166$, approximately 64K multiplications and 32K additions must be performed. In a 30 MFLOPS processor this would require about $3.2msec$.

6.2. CAC Algorithm for Type A (Loss-free) Service

Assume that source i requests the provision of loss-free transmission (Type A Service). One approach to providing this service is to reserve the requested bandwidth $\check{r}_{H_i}$ for the whole duration of the call. The decision algorithm in this case is the following: if the sum of the rates $\check{r}_{H_i}$ of all streams requesting loss-free transmission (Type A service) exceeds the link rate $\check{R}_l$, reject the call; otherwise rerun the CAC algorithms for all other sources under the new bandwidth $\check{R}_l - \check{r}^{lf}_{TOT}$, where $\check{r}^{lf}_{TOT}$ is the cumulative rate of all loss-free streams (Type A sources) expressed in cells/frame. Such an approach, while commonly proposed for this type of service, leads to underutilization of network resources. In order to increase the utilization of the network resources, the solution proposed in this paper utilizes the statistical characteristics of the loss-free streams.

The CAC Algorithm for Type A sources is the following:

Step 1: Check if the cumulative rate of Type A sources is less than or equal to the link rate ($\check{r}^{lf}_{TOT} \leq \check{R}_l$). If not, reject the incoming call; otherwise go to Step 2.

Step 2: Check if all the QoS requirements can still be satisfied for the already accepted sources, under the presence of the new loss-free connection. For this purpose, the Type A (requesting loss-free service) VCs are considered as being Type B (requesting burst-scale service) VCs. Update the coefficients $\{C^b_j\}$, and rerun the algorithms for all VCs except those requesting Type A service.

The reason Step 2 can be performed without having to change the connection admission control algorithms is that the burst-loss probability predicted is an upper bound that does not take into account the order in which the sources arrive, but considers that the source under consideration is always among those that lose their bursts in case of congestion. To explain this further, let us examine the meaning of $Pr(X - x_i > \check{R}_l - \check{r}_{H_i})$. Since its computation is based on the coefficients C^b_j, which represent the probability that the total rate of all active sources at a random instant is j slots/frame, $Pr(X - x_i > \check{R}_l - \check{r}_{H_i})$ shows only the probability that the rate of all active sources other than source i exceeds $\check{R}_l - \check{r}_{H_i}$. This does not necessarily mean that the burst of source i is among the bursts that are dropped due to lack of available bandwidth. Even if $X - x_i > \check{R}_l - \check{r}_{H_i}$, the burst of source i can be among the ones accepted in Q_1. Thus the upper bound is also valid in the presence of loss-free traffic, the bursts of which will always be among those admitted.

For the cells of Type C VCs, there is no distinction between bursts of Type A and Type B VCs, since they both have the same effect on the cell-loss probability of Type C VCs.

6.3. CAC Algorithm for Type C (Cell-scale) Service

Let i be the source under consideration and

$$C^c_j = Prob\{\text{cumulative rate of all active cell-scale sources except i in cells/slot} = j\}$$

Since the cell-scale sources can have one of two rates r_H or r_L when active, then the new coefficients $\{\hat{C}^c_j\}$ when source i is connected can be obtained by:

$$\hat{C}^c_j = C^c_j \cdot \bar{p}_i + C^c_{j-\check{r}_{L_i}} \cdot p_i \cdot p_{L_i} + C^c_{j-\check{r}_{H_i}} \cdot p_i \cdot p_{H_i} \tag{6}$$

for all $j \geq 0$. Similarly, the coefficients $\{\hat{C}_j^c\}$ when source i is disconnected can be computed using the equations:

$$\hat{C}_j^c = \frac{1}{\bar{p}_i} \cdot (C_j^c - \hat{C}_{j-\check{r}_{L_i}}^c \cdot p_i \cdot p_{L_i} - \hat{C}_{j-\check{r}_{H_i}}^c \cdot p_i \cdot p_{H_i}) \tag{7}$$

where all coefficients with negative subscripts are considered to be zero in the above equations.

Let $\check{r}_{TOT}^c$ be the sum of the rates of all the accepted VCs of Type C in slots/frame. Similarly, let $\check{r}_{TOT}^b$ be the sum of the rates of all the accepted VCs of Type A and Type B in slots/frame. Let $C_{over}^b \stackrel{\text{def}}{=} \sum_{j=\check{R}_l}^{\check{r}_{TOT}^b} C_j^b$ denote the probability that the total rate of all active VCs requesting Type A and Type B service at any given moment is greater than or equal to the available bandwidth. C_{over}^b is an upper bound on the probability that the Type A and Type B VCs have occupied all the available bandwidth. It is further assumed that the number of lost cells of a VC requesting Type C service is proportional to its rate. An upper bound on the cell loss probability for source i is given by:

$$PL_i = \sum_{m=0}^{\check{R}_l-1} C_m^b \cdot \Big(\sum_{j=\check{R}_l-m-\check{r}_{L_i}}^{min(\check{r}_{TOT}^c, 4\cdot\check{R}_l)} \frac{N_{drop_{L_i}}}{\check{r}_{L_i}} \cdot p_{L_i} \cdot C_j^c + \sum_{j=\check{R}_l-m-\check{r}_{H_i}}^{min(\check{r}_{TOT}^c, 4\cdot\check{R}_l)} \frac{N_{drop_{H_i}}}{\check{r}_{H_i}} \cdot p_{H_i} \cdot C_j^c \Big) + C_{over}^b$$

$$= \frac{p_{L_i}}{\check{r}_{L_i}} \cdot \sum_{m=0}^{\check{R}_l-1} C_m^b \cdot \Big(\sum_{j=\check{R}_l-m-\check{r}_{L_i}}^{min(\check{r}_{TOT}^c, 4\cdot\check{R}_l)} N_{drop_{L_i}} \cdot C_j^c \Big) + \frac{p_{H_i}}{\check{r}_{H_i}} \cdot \sum_{m=0}^{\check{R}_l-1} C_m^b \cdot \Big(\sum_{j=\check{R}_l-m-\check{r}_{H_i}}^{min(\check{r}_{TOT}^c, 4\cdot\check{R}_l)} N_{drop_{H_i}} \cdot C_j^c \Big) + C_{over}^b \tag{8}$$

where

$$N_{drop_{L_i}} = (j + \check{r}_{L_i} - (\check{R}_l - m)) \cdot \frac{\check{r}_{L_i}}{j + \check{r}_{L_i}} \;=\; (j + m - \check{R}_l + \check{r}_{L_i}) \cdot \frac{1}{j/\check{r}_{L_i} + 1}$$

$$and \quad N_{drop_{H_i}} \;=\; (j + m - \check{R}_l + \check{r}_{H_i}) \cdot \frac{1}{j/\check{r}_{H_i} + 1}$$

The time required to check if a new source can be accepted can be computed as follows. Store the coefficients representing the distribution of the cumulative rate (in slots/frame) of all active cell-scale sources including the source requesting connection. Then to decide whether source i received the requested QoS in the presence of the source requesting acceptance, the coefficients $\{C_j^c\}$ can be computed from the stored distribution of the cumulative rate by removing source i. If $j_{max} \stackrel{\text{def}}{=} 4 \cdot \check{R}_l$, then each time we update the coefficients we need to perform $3 \cdot j_{max}$ multiplications and $2 \cdot j_{max}$ additions. Assuming that $1/\check{r}_{L_i}$ and $(\check{R}_l - \check{r}_{H_i})$ are stored in memory, each of the internal summations requires 3 additions, 2 multiplications and 1 division, and is performed in at most

$$\sum_{m=0}^{\check{R}_l-1} (4 \cdot \check{R}_l - \check{R}_l + \check{r} + m) \;=\; \check{R}_l \cdot (3 \cdot \check{R}_l + \check{r}) + \frac{(\check{R}_l - 1) \cdot \check{R}_l}{2}$$

operations, where $r = r_{L_i}$ for the first summation in (8) and $r = r_{H_i}$ for the second. Thus, the time complexity of the CAC algorithm for a source requesting Type C service is approximately given by:

$$A_1 \cdot n \cdot 4 \cdot \check{R}_l + A_2 \cdot \sum_{i=1}^{n} \Big(\check{R}_l \cdot (6 \cdot \check{R}_l + \check{r}_{L_i} + \check{r}_{H_i}) + (\check{R}_l - 1) \cdot \check{R}_l \Big) \tag{9}$$

where n is the total number of VCs requesting service of Type C, $(A_1, A_2) = (2, 4)$ for the number of additions, $(A_1, A_2) = (3, 2)$ for the number of multiplications and $(A_1, A_2) = (0, 1)$ for the number of divisions.

Note that the above number of operations is an upper bound due to the assumption that $\check{r}^c_{TOT} \geq 4 \cdot \check{R}_l$. Also note the distinction between divisions and other operations. This is done because divisions need to be multiplied by a factor of 4, in order to use the peak rate MFLOPS ratings of a microprocessor [9].

As a example, consider the case of $n = 200$ VCs requesting service of Type C, and for simplicity assume that all sources have the same peak rate $\lambda = 1Mbps$. If $R_l = 155Mbps$, $T = .456msec$ (corresponding to 166 cells per frame and a buffer size $L = 3 \cdot 166 \approx 500cells$), $r_{step} = .93Mbps$, $r_{L_i} = .93Mbps$, $r_{H_i} = 1.86Mbps$, $\hat{R}_l = 154.38Mbps$, the computations would require approximately $12.9sec$ on a 30 MFLOPS (peak rate) processor. The $12.9sec$ is an excessive time for a CAC algorithm. Given that, with the introduction of r_L and r_H, the inefficiency introduced by small T (or large r_{step}) has been partially eliminated, T can be reduced in order to reduce the decision time, without introducing excessive inefficiency. For example if $T = 91.18\mu sec$ (corresponding to 33 cells per frame and a buffer size $L = 3 \cdot 33 \approx 100cells$), or $r_{step} = 4.65Mbps$, $r_L = 0$ $r_H = r_{step}$, then the time needed is approximately $510msec$ on a 30 MFLOPS (peak) processor. The time may still be considered excessive, in which case either a smaller frame size T or the upper bounding approximation presented in Section 7 must be used.

7. Reducing the Complexity for the Cell Level CAC Algorithm

The algorithm presented in the previous section for the computation of the quality of service for the cell-scale sources has two problems:

1. It requires excessive computation time.
2. The time required increases almost linearly with the number of sources.

In order to take care of both of these problems, an upper bounding approximate solution with the following properties has been developed:

1. The solution reduces dramatically the run time of the algorithm.
2. The number of sources has an almost negligible effect on the run time.

Let us redefine the coefficients C_j^c as:

$$C_j^c = Prob\{\text{cumulative rate of all active cell-scale sources in cells/slot}= j\}$$

The equations for updating C_j^c are still given by equations (6) and (7). Let

$$S_j^c \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l} C_i^c/i \; = \; S_{j+1}^c + C_j^c/j$$

$$\check{S}_j^c \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l} C_i^c \; = \; S_{j+1}^c + C_j^c$$

$$\hat{S}_j^c \stackrel{\text{def}}{=} \sum_{i=j}^{\check{R}_l-1} S_i^c \; = \; \hat{S}_{j+1}^c + S_j^c$$

$$S_j^b \stackrel{\text{def}}{=} \sum_{i=j}^{\check{R}_l-1} C_i^b \; = \; S_{j+1}^b + C_j^b$$

$$C_{over}^b \stackrel{\text{def}}{=} \sum_{i=\check{R}_l}^{4\cdot\check{R}_l} C_i^b = 1 - S_0^b$$

Then, performing some upper bound approximations and some algebraic manipulations, the calculation of the cell-loss probability is given by the following equation:

$$LP_i \; = \; p_{L_i} \cdot \check{F}(\check{r}_{L_i}) + p_{H_i} \cdot \check{F}(\check{r}_{H_i}) + C_{over}^b$$

where

$$\check{F}(0) \; = \; 0 \quad ,$$

$$\check{F}(1) \; = \; \frac{S_0^b}{\check{R}_l} \cdot \Big(\sum_{i=\check{R}_l-1}^{4\cdot\check{R}_l} i \cdot C_i^c - (\check{R}_l - 1) \cdot \check{S}_{\check{R}_l-1}^c \Big) + \sum_{m=1}^{\check{R}_l-1} \hat{S}_{\check{R}_l-m-1}^c \cdot C_m^b - \hat{S}_{\check{R}_l-1}^c \cdot S_1^b \quad ,$$

and

$$\check{F}(\check{x}) \; = \; \check{F}(\check{x}-1) + \frac{\check{S}_{\check{R}_l-\check{x}+1}^c \cdot S_0^b}{\check{R}_l} + \sum_{m=1}^{\check{R}_l-1} S_{\check{R}_l-m-\check{x}}^c \cdot C_m^b - S_{\check{R}_l-\check{x}}^c \cdot S_1^b \quad ,$$

for $\check{x} \geq 1$.

If n is the number of cell-scale sources, then the total number of calculations for this algorithm is given by:

$$4 \cdot n + 2 \cdot (\check{R}_l - 1)^2 + 52 \cdot \check{R}_l + 8 \cdot (\check{R}_l - 1) + 9 \quad . \tag{10}$$

The details of this derivation are presented in the appendix.

Figure 3 illustrates the effect of the upper bounding approximation on the computation of the cell-loss probability, under the presence of a number of burst scale sources, with the same traffic characteristics as the cell-scale sources, and for a variety of peak and mean values. All plots refer to the case $R_l = 155Mbps$ and frame size $T = .456\mu sec$ (corresponding to 166 cells per frame). Clearly the approximations have a very small impact on the accuracy of the computed loss probability and get better as the number of sources increases.

In the previous section, it was shown that the frame size had a dramatic effect on the running time of the algorithm for the exact solution. Figure 4 illustrates the effect of the frame size when the approximate solution is used for the calculation of the cell-loss probability, for the case where a mix of sources is present. The plot presents the *worst-case running time,* which corresponds to the case in which either a loss-free or a burst-scale source requests connection; in this case it is necessary to calculate the loss probability for all the already accepted sources. The link rate for the plot is assumed to be 155Mbps.

Finally, Figure 5 illustrates the effect of the number of sources on the time needed for the decision concerning the acceptance of a new source. The traffic is a mix of an equal number of burst-scale and cell-scale sources. The horizontal axis shows the total number of sources. The plots show the running time for the burst-scale and the cell-scale algorithms along with the sum of the two times. It is clearly seen that the time for the cell-scale algorithm remains practically constant for a large range of values of the number of sources.

8. Size of the Frame

The size of the frame is a very important parameter for the system operation. As mentioned earlier, the choice of frame size is dictated by the available buffer size and the limits imposed on the jitter and end-to-end delay. It is clear that reducing the frame size leads to smaller buffer requirements, smaller end-to-end delay and stricter jitter bounds. Moreover, as can be seen from Equations 9, 10, it drastically reduces the running time of the cell-scale admission control algorithm since the values of $\check{R}_l, \check{r}_{H_i}, \check{r}_{L_i}$ are proportional to T. On the other hand, the inverse proportional relationship between the frame size T and the bandwidth allocation step r_{step} increases (except in certain cases which are examined later on) the inefficiency introduced by the discrete bandwidth allocation when the frame size is reduced. Thus, the frame size must be a compromise between a large frame leading to efficient utilization and a small frame providing all the other desirable properties, but introducing bandwidth inefficiency.

Figure 6 shows the burst-loss probability for burst-scale sources for different frame sizes and for a variety of traffic characteristics. The link rate R_l is $155Mbps$. The impact of the frame size and the inefficiency introduced by small frame sizes can clearly be seen. Note that in the figures with peak rate $10Mbps$, the curves coincide for frame sizes equal to $.45msec$ and $.9msec$ although one would expect the bigger frame size ($.9msec$) to result in lower loss probabilities. The curves coincide because in both cases the (homogeneous) sources are mapped on the same rate r_H ($r_H(T = .45msec) = \lceil 4.56 \cdot 10^{-4} \cdot 10^7/384 \rceil \cdot .93Mbps = 11.16Mbps$ and $r_H(T = .9msec) = \lceil 9.12 \cdot 10^{-4} \cdot 10^7/384 \rceil \cdot .465Mbps = 11.16Mbps$). In the figure with peak rate $1Mbps$ an even stronger effect takes place and a smaller frame size $T = .27msec$ leads to lower loss probabilities than the larger frame size $T = .45msec$. This is because in this particular case with homogeneous sources, the smaller frame gives a closer mapping of the rate of the source to the rate r_H assigned to it. In this example

$r(T = .27msec) = 1 \cdot r_{step} = 1.55Mbps$ while $r(T = .45msec) = 2 \cdot r_{step} = 1.86msec$. Such strange situations are unlikely to appear in an actual network.

The two-rate mapping $((r_L, p_L), (r_H, p_H))$ introduced to describe the cell-scale streams partially eliminates the effects of the frame size on the efficiency of the system. Figure 7 shows the cell-loss probability for cell-scale sources for different frame sizes and for a variety of traffic characteristics. The link rate R_l is $155Mbps$. It can clearly be seen that the inefficiency introduced by smaller frame sizes is reduced due to the two-rate mapping used.

The conclusion from Figures 6 and 7 is that a frame size in the range $[270, 450]\mu sec$, corresponding to a buffer size in the range $[300, 500]cells$ and a maximum jitter in the range $[540, 900]\mu sec$ for an 155Mbps link rate provides a good efficiency in all the cases examined, implying that this range is an appropriate interval of frame sizes for the operation of the system.

9. Policing of Source Characteristics

The decision to accept or reject a call is based on the traffic characteristics that the source declares at the call setup time. In order to guarantee that the performance seen by all conforming sources is not affected by the presence of non-conforming sources, it is necessary to police individually the operation of each source, making sure that it does not violate the declared parameters.

The parameters declared by the user at the connection setup time are the mean rate and the peak rate; thus it would be sufficient to provide policing mechanisms for these two parameters. The mean rate can be policed through the use of a leaky bucket proposed in [14] at the input to the network. For policing the peak rate, a cell-spacer proposed in [3] can be used. Note that because of the framing strategy, the rates r, r_L and r_H will not be violated as the stream goes through the network, so it is sufficient to police the peak rate at the entrance to the network.

The parameters used in the admission control algorithms are the probability that a source is active and the maximum rate $\check{r}_H$ within each frame ((r_L, r_H) in the case of cell-scale sources). Instead of policing the peak rate, it is sufficient to impose the conformance of the source to these parameters. In this way the source is permitted more freedom regarding how it generates cells. Policing the mean rate can still be performed through the use of a leaky bucket mechanism at the edge of the network. Even though violation cannot occur in internal nodes under normal operating conditions (no faulty nodes), we propose very simple policing mechanisms that can be applied at every node to protect its operation from previous malfunctioning nodes. The remaining of this section presents policing mechanisms which are much simpler than the cell-spacer ([3]), since they consist of a single counter per source.

Loss-free and burst-scale sources

Traffic from a loss-free or burst-scale source should provide at most $\check{H}_r$ cells within each time frame T. Policing at the edge of the network is achieved by a counter b_i, for each source i. Initially b_i is set to $\check{r}_{H_i} - 1$. The counter b_i is decremented every time a cell of stream i arrives. When b_i becomes equal to zero, then all subsequent arriving cells of stream i are dropped and b_i remains unchanged. Counter b_i is reset to $\check{r}_{H_i} - 1$ at the beginning of each frame.

Since the mechanism presented above is very simple, exactly the same mechanism can optionally be used for policing the sources at every node, in order to protect its operation from previous malfunctioning nodes.

Cell-scale sources

In the case of cell-scale sources, the source is assumed to have two rates $\check{r}_L$ and $\check{r}_H$, with probabilities p_L and p_H respectively. Exactly the same policing mechanism exercised in the case of loss-free and burst-scale sources can also be applied here by setting the counter equal to $\check{r}_H$. The enforcement of the ratio of frames containing a low and a high number of cells (specifying p_L and p_H) is guaranteed by the policing of the mean rate of the source through a leaky bucket. The same single-counter mechanism can optionally be applied at each node of the network for protection purposes.

10. Specification of Q_2

Q_2 provides guarantees of cell-level loss-probability, but no guarantees for the delay and the jitter of the sources. Cells from Q_2 are serviced only when no cell from Q_1 is eligible for service. This restriction guarantees that cells from Q_2 will not interfere with the service provided to cells in Q_1.

As has already been seen in the specification of the operation of Q_1, the server for Q_1 can stay idle even when Q_1 is not empty (corresponding to the case where all cells from the current outgoing frame have been serviced and the server remains idle until the beginning of the next frame). This operation of the server belongs to a general family of servers called non-work-conserving. A study of such servers can be found in [16]. Q_2 partially eliminates the inefficiency introduced by the non-work-conserving server Q_1 by using, whenever not empty, the idle periods of Q_1.

10.1. Running Time Specification

Incoming cells are buffered if Q_2 is not full and dropped otherwise. Whenever outgoing frames have empty slots, cells from Q_2 are serviced.

10.2. Connection Admission Control Algorithm

The probability that a slot in the output link is empty will provide an approximation for the residual bandwidth provided to Q_2. Assume the coefficients $C_m^{(1)}$ have been computed using the admission control algorithm for the sources in Q_1, where $C_m^{(1)}$ is the probability that the bandwidth required at a random instant by the sources in Q_1 is $m \cdot r_{step}$. The coefficients

$C_m^{(1)}$ are easily computed by using equations (6) and (7) and including all loss-free and burst-scale sources in these equations by assigning $(r_L, r_H) = (0, r_H)$, and $(p_L, p_H) = (0, 1)$.

Given that the number of cells that can be served per frame is $\check{R}_l$, then with probability $C_m^{(1)}$, $\check{R}_l - m$ slots per frame are free. The average number of free cells per frame is given by:

$$\sum_{m=0}^{\check{R}_l} C_m^{(1)} \cdot (\check{R}_l - i)$$

which approximately corresponds to a rate:

$$R_l^{(2)} = r_{step} \cdot \sum_{m=0}^{\check{R}_l} C_m^{(1)} \cdot (\check{R}_l - i) \quad .$$

$R_l^{(2)}$ signifies the available link rate to cells from queue Q_2. Due to the discretization of bandwidth allocation in multiples of r_{step}, the users observe a link rate $\hat{R}_l$ which is equal to

$$\hat{R}_l^{(2)} = \lfloor \frac{R_l^{(2)}}{r_{step}} \rfloor \cdot r_{step}$$

In order to meet the real time decision requirements, a *bufferless model* [6] can be used. With the bufferless model, whenever the instantaneous cumulative rate exceeds the link rate, the excess rate is lost. On the other hand, the buffer can accommodate any contention due to simultaneous arrivals of cells from different streams. By using the bufferless model, an upper bound to the loss probability provided to source i cells can be computed.

Imposing no restriction on the values of the peak rate of the sources can lead to an extremely large number of coefficients $C_j^{(2)}$, the computation of which cannot be performed by a real-time decision algorithm.

To simplify the analysis, it is necessary to restrict the peak rate of each source of Class D to a multiple of some basic rate, called $r_{step}^{(2)}$. Source j is then assigned a rate $\hat{\lambda}_j = \lceil \frac{\lambda_j}{r_{step}^{(2)}} \rceil \cdot r_{step}^{(2)}$. To simplify the notation, let

$$\tilde{R}_l^{(2)} \stackrel{\text{def}}{=} \frac{\hat{R}_l^{(2)}}{r_{step}^{(2)}} \quad .$$

Let x_j be the random variable showing the amount of bandwidth needed by source j at a random instant. Then $Pr(x_j = \hat{\lambda}_j) = \mu_j/\lambda_j = p_j$ and $Pr(x_j = 0) = 1 - \mu_j/\lambda_j = \bar{p}_j$.

Also let $X \stackrel{\text{def}}{=} \sum_{j=1}^{N} x_j$ and

$$C_j^{(2)} \stackrel{\text{def}}{=} Prob(X = j \cdot r_{step}^{(2)})$$

The updating of the coefficients $C_j^{(2)}$ when a source is connected or disconnected is performed in exactly the same way as in the case of C_j^b (equations (3) and (4)). Once again it is assumed that all coefficients with $j > 4 \cdot \check{R}_l^{(2)}$ can be neglected.

Let $R_l^{(2)}$ be the service rate provided to queue Q_2, and let $\check{R}_l \stackrel{\text{def}}{=} \lfloor R_l / r_{step}^{(2)} \rfloor$, $\check{R}_l^{(2)} \stackrel{\text{def}}{=} R_l^{(2)} / r_{step}^{(2)}$, and $\check{\lambda} \stackrel{\text{def}}{=} \lambda / r_{step}^{(2)}$. Then the loss probability for source i, as given by the bufferless model, is:

$$LP_i = p_i \cdot \sum_{j=\check{R}_l^{(2)}-\check{\lambda}_i}^{4\cdot\check{R}_l} C_j^{(2)} \cdot \left(1 - \frac{\check{R}_l^{(2)}}{j + \check{\lambda}_i}\right)$$

Define:

$$S_j^{Q_2} \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l} C_i^{(2)} = S_{j+1}^{Q_2} + C_j^{(2)} \quad ,$$

and

$$\check{S}_j^{Q_2} \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l} \frac{C_i^{(2)}}{i + \check{R}_l} = \check{S}_{j+1}^{Q_2} + \frac{C_j^{(2)}}{j + \check{R}_l} \quad .$$

Then,

$$LP_i = p_i \cdot \left(S_{\check{R}_l^{(2)}-\check{\lambda}_i}^{Q_2} - \check{R}_l^{(2)} \cdot \check{S}_{\check{\lambda}_i}^{Q_2}\right)$$

Consider now the algorithm running time. The number of instructions required is given by:

$C_j^{(2)} \rightarrow 4 \cdot \check{R}_l \cdot (1add + 2mul)$

$S_j^{Q_2} \rightarrow 4 \cdot \check{R}_l \; add$

$\check{S}_j^{Q_2} \rightarrow 4 \cdot \check{R}_l \cdot (2add + 1div)$

$LP \rightarrow n \cdot (1add + 2mul)$

$\check{R}_l^{(2)} \rightarrow 4 \cdot \frac{R_l}{r_{step}} \cdot (2add + 3mul) \; + \; (\check{R}_l + 1) \cdot (2add + 1mul)$

Adding the above terms, the total number of calculations is:

$$63 \cdot \check{R}_l + 3 \cdot n + 3 \quad .$$

For n=200 sources with peak rate $\lambda = 1Mbps$ each, $R_l = 155Mbps$ and $r_{step}^{(2)} = 500Kbps$, the computation would require approximately $691\mu sec$ in a 30 MFLOPS processor. Reducing this time can easily be achieved by increasing $r_{step}^{(2)}$. For example for $r_{step}^{(2)} = 2Mbps$, the computation would require approximately $184\mu sec$.

Figure 8 shows the worst-case running time requirements (corresponding to the case when a loss-free or a burst-scale source requests connection), for the case when $R_l = 155Mbps$ and the step of bandwidth allocation for the traffic entering Q_2 is $r_{step}^{(2)} = 1Mbps$.

If Q_2 were infinite, then the service rate that cells of Q_2 would see would be exactly equal to $R_l^{(2)}$ in the limit of an infinite observation interval. The fact that the empty slots are not regularly spaced, coupled with the finite buffer Q_2, is what causes $R_l^{(2)}$ to be just an approximation which improves as Q_2 increases. The approximation introduced by the way of computing $R_l^{(2)}$ is the reason for calling the guarantees provided by Q_2 *soft guarantees* .

11. Conclusions

This paper has presented and analyzed a traffic management scheme in which sources requiring guarantees at the burst level can coexist with sources requiring guarantees at the cell-level and sources demanding loss-free transmission. The novel feature of the loss-free service is that the idle periods of the loss-free sources are taken into account in the computation of the loss probability for the burst-scale and the cell-scale sources.

Delay and jitter are strictly bounded for the VCs requesting Types A, B and C type of service. Type D service provides soft guarantees at the cell-loss probability. All four types of services can be supported at the expense of the addition of a small hardware complexity at each switch. For each type of service, the running time actions the switch should take upon the arrival of a cell have been presented. The connection admission control algorithms presented are real-time. In all the examples tried and for a large number of sources (of the order of 1000), the run time has always been less than $10msec$ for a $30MFLOPS$ processor. A typical mix of 500 sources requires less than $5msec$ in most practical cases.

Future work will focus primarily on the provision of different losses to sources with different loss requirements, through the introduction of a set of priority levels.

12. Appendix

Reducing the complexity of the cell-scale algorithm

The equation for the computation of the cell-loss probability is given by:

$$PL_i = \frac{p_{L_i}}{\check{r}_{L_i}} \cdot \sum_{m=0}^{\check{R}_l-1} C_m^b \cdot \sum_{j=\check{R}_l-m-\check{r}_{L_i}}^{4\cdot\check{R}_l} N_{drop_{L_i}} \cdot C_j^c + \frac{p_{H_i}}{\check{r}_{H_i}} \cdot \sum_{m=0}^{\check{R}_l-1} C_m^b \cdot \sum_{j=\check{R}_l-m-\check{r}_{H_i}}^{4\cdot\check{R}_l} N_{drop_{H_i}} \cdot C_j^c + C_{over}^b$$

where

$$N_{drop_{L_i}} = (j + m - \check{R}_l + \check{r}_{L_i}) \cdot \frac{1}{j/\check{r}_{L_i} + 1}$$

and similarly for $N_{drop_{H_i}}$.

Define:

$$N_{drop}(j, m, \check{r}) \stackrel{\text{def}}{=} (j + m - \check{R}_l + \check{r}) \cdot \frac{1}{j/\check{r} + 1}$$

and

$$A(m,\check{r}) \stackrel{\text{def}}{=} \sum_{j=\check{R}_l-m-\check{r}}^{4\cdot\check{R}_l} N_{drop}(j,m,\check{r})\cdot C_j^c$$

Then,

$$N_{drop}(j,m+1,\check{r}) \;=\; (j+(m+1)-\check{R}_l+\check{r})\cdot\frac{1}{j/\check{r}+1} \;=\; N_{drop}(j,m,\check{r})+\frac{1}{j/\check{r}+1}$$

Also,

$$A(m+1,\check{r}) \;=\; \sum_{j=\check{R}_l-m-\check{r}-1}^{4\cdot\check{R}_l} N_{drop}(j,m+1,\check{r})\cdot C_j^c$$

$$= \sum_{j=\check{R}_l-m-\check{r}}^{4\cdot\check{R}_l} N_{drop}(j,m,\check{r})\cdot C_j^c + \sum_{j=\check{R}_l-m-\check{r}-1}^{4\cdot\check{R}_l}\frac{C_j^c}{j/\check{r}+1}$$

$$\leq\; A(m,\check{r})+\check{r}\cdot\sum_{j=\check{R}_l-m-\check{r}-1}^{4\cdot\check{R}_l}\frac{C_j^c}{j}\quad. \tag{11}$$

Define now:

$$S_j^c \stackrel{\text{def}}{=} \sum_{\mathrm{i}=\mathrm{j}}^{4\cdot\check{\mathrm{R}}_\mathrm{l}} \mathrm{C}_\mathrm{i}^\mathrm{c}/\mathrm{i} \;=\; \mathrm{S}_{\mathrm{j}+1}^\mathrm{c}+\mathrm{C}_\mathrm{j}^\mathrm{c}/\mathrm{j}$$

Then (11) becomes:

$$A(m+1,\check{r}) \;\leq\; A(m,\check{r})+\check{r}\cdot S^c_{\check{R}_l-m-\check{r}-1}$$

$$=\; A(m-1,\check{r})+\check{r}\cdot(S^c_{\check{R}_l-m-\check{r}-1}+S^c_{\check{R}_l-m-\check{r}}) \;=\; \ldots \;=\; A(0,\check{r})+\check{r}\cdot\sum_{\check{R}_l-m-\check{r}-1}^{\check{R}_l-\check{r}-1} S_j^c$$

Define:

$$\hat{S}_j^c \stackrel{\text{def}}{=} \sum_{\mathrm{i}=\mathrm{j}}^{\check{\mathrm{R}}_\mathrm{l}-1} \mathrm{S}_\mathrm{i}^\mathrm{c} \;=\; \hat{\mathrm{S}}_{\mathrm{j}+1}^\mathrm{c}+\mathrm{S}_\mathrm{j}^\mathrm{c}$$

Then,

$$A(m,\check{r}) \;=\; A(0,\check{r})+\check{r}\cdot(\hat{S}^c_{\check{R}_l-m-\check{r}}-\hat{S}^c_{\check{R}_l-\check{r}}) \tag{12}$$

Specify $A(0,\check{r})$ as follows:

$$A(0,\check{r}) \;=\; \sum_{j=\check{R}_l-\check{r}}^{4\cdot\check{R}_l}(j-\check{R}_l+\check{r})\cdot\frac{1}{j/\check{r}+1}\cdot C_j^c$$

Let

$$G(j,\check{r}) \stackrel{\text{def}}{=} (j-\check{R}_l+\check{r})\cdot\frac{1}{j/\check{r}+1}$$

Then,

$$G(j+1,\check{r}) \;=\; (j+1-\check{R}_l+\check{r})\cdot\frac{1}{(j+1)/\check{r}+1} \;<\; G(j,\check{r})+\frac{1}{(j+1)/\check{r}+1}$$

$$<\; G(j,\check{r})+\frac{1}{\frac{\check{R}_l-\check{r}}{\check{r}}+1} \;=\; G(j,\check{r})+\frac{\check{r}}{\check{R}_l}$$

and

$$A(0,\check{r}) \;<\; \sum_{j=\check{R}_l-\check{r}}^{4\cdot\check{R}_l}(G(j-1,\check{r})+\frac{\check{r}}{\check{R}_l})\cdot C_j^b \;=\; \sum_{j=\check{R}_l-\check{r}}^{4\cdot\check{R}_l}(G(\check{R}_l-\check{r},\check{r})+(j-(\check{R}_l-\check{r}))\cdot\frac{\check{r}}{\check{R}_l})\cdot C_j^b$$

Let

$$\check{S}_j^c \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l}C_i^c \;=\; \check{S}_{j+1}^c+C_j^c$$

and

$$\ddot{S}_j^c \stackrel{\text{def}}{=} \sum_{i=j}^{4\cdot\check{R}_l}i\cdot C_i^c \;=\; \ddot{S}_{j+1}^c+j\cdot C_j^c$$

Then,

$$A(0,\check{r}) \;<\; (G(\check{R}_l-\check{r},\check{r})-\frac{\check{r}}{\check{R}_l}\cdot(\check{R}_l-\check{r}))\cdot\check{S}_{\check{R}_l-\check{r}}^c+\frac{\check{r}}{\check{R}_l}\cdot\ddot{S}_{\check{R}_l-\check{r}}^c$$

Note that

$$G(\check{R}_l-\check{r},\check{r}) \;=\; (\check{R}_l-\check{r}-(\check{R}_l-\check{r}))\cdot\frac{1}{\frac{\check{R}_l-\check{r}}{\check{r}}+1} \;=\; 0$$

which gives

$$A(0,\check{r}) \;<\; \frac{\check{r}}{\check{R}_l}\cdot(\ddot{S}_{\check{R}_l-\check{r}}^c-(\check{R}_l-\check{r})\cdot\check{S}_{\check{R}_l-\check{r}}^c)$$

From now on the right hand side of the above inequality will be used for the computation of $A(0,\check{R}_l)$.

$$A(0,\check{r}+1) \;=\; \frac{\check{r}+1}{\check{R}_l}\cdot(\ddot{S}_{\check{R}_l-\check{r}-1}^c-(\check{R}_l-\check{r}-1)\cdot\check{S}_{\check{R}_l-\check{r}-1}^c)$$

$$=\; \frac{\check{r}+1}{\check{R}_l}\cdot(\ddot{S}_{\check{R}_l-\check{r}}^c+(\check{R}_l-\check{r}-1)\cdot C_{\check{R}_l-\check{r}-1}^c-(\check{R}_l-\check{r}-1)\cdot(\check{S}_{\check{R}_l-\check{r}}^c+C_{\check{R}_l-\check{r}-1}^c))$$

$$=\; \frac{\check{r}+1}{\check{r}}\cdot(\ddot{S}_{\check{R}_l-\check{r}}^c-(\check{R}_l-\check{r})\cdot\check{S}_{\check{R}_l-\check{r}}^c)+\frac{\check{r}+1}{\check{R}_l}\cdot\check{S}_{\check{R}_l-\check{r}}^c$$

$$=\; \frac{\check{r}+1}{\check{r}}\cdot A(0,\check{r})+\frac{\check{r}+1}{\check{R}_l}\cdot\check{S}_{\check{R}_l-\check{r}}^c \qquad \text{for } \check{r}>0$$

Let us define:

$$F(\check{r}) \stackrel{\text{def}}{=} \sum_{m=0}^{\check{R}_l-1}C_m^b\cdot\sum_{j=\check{R}_l-m-\check{r}}^{4\cdot\check{R}_l}N_{drop}(j,m,\check{r})\cdot C_j^c \;=\; \sum_{m=0}^{\check{R}_l-1}A(m,\check{r})\cdot C_m^b$$

With the introduction of the new C_j^c coefficients (including the contribution of the source under consideration), it is easy to see that F depends only on the rate of the source. Thus:

$$PL_i \;=\; \frac{p_{L_i}}{\check{r}_{L_i}} \cdot F(\check{r}_{L_i}) + \frac{p_{H_i}}{\check{r}_{H_i}} \cdot F(\check{r}_{H_i}) + C^b_{over} \;=\; p_{L_i} \cdot \check{F}(\check{r}_{L_i}) + p_{H_i} \cdot \check{F}(\check{r}_{H_i}) + C^b_{over}$$

where

$$\check{F}(\check{r}) = F(\check{r})/\check{r}$$

Introducing (12) in the definition of $F(\check{r})$,

$$F(\check{r}) \;=\; A(0,\check{r}) \cdot \sum_{m=0}^{\check{R}_l-1} C^b_m + \check{r} \cdot \sum_{m=1}^{\check{R}_l-1} \hat{S}^c_{\check{R}_l-m-\check{r}} \cdot C^b_m - \check{r} \cdot \sum_{m=1}^{\check{R}_l-1} \hat{S}^c_{\check{R}_l-\check{r}} \cdot C^b_m$$

Let

$$S^b_j \;=\; \sum_{i=j}^{\check{R}_l-1} C^b_i \;=\; S^b_{j+1} + C^b_j$$

Then

$$F(\check{r}) \;=\; A(0,\check{r}) \cdot S^b_0 + \check{r} \cdot \sum_{m=1}^{\check{R}_l-1} \hat{S}^c_{\check{R}_l-m-\check{r}} \cdot C^b_m - \check{r} \cdot \hat{S}^c_{\check{R}_l-\check{r}} \cdot S^b_1$$

and

$$F(\check{r}+1) \;=\; A(0,\check{r}+1) \cdot S^b_0 + (\check{r}+1) \cdot \sum_{m=1}^{\check{R}_l-1} \hat{S}^c_{\check{R}_l-m-\check{r}-1} \cdot C^b_m - (\check{r}+1) \cdot \hat{S}^c_{\check{R}_l-\check{r}-1} \cdot S^b_1$$

which for $\check{r} \geq 1$ can be written as:

$$\begin{aligned}
F(\check{r}+1) \;&=\; (\frac{\check{r}+1}{\check{r}} \cdot A(0,\check{r}) + \frac{\check{r}+1}{\check{R}_l} \cdot \check{S}^c_{\check{R}_l-\check{r}}) \cdot S^b_0 + \\
&+(\check{r}+1) \cdot \sum_{m=1}^{\check{R}_l-1} (\hat{S}^c_{\check{R}_l-m-\check{r}} + S^c_{\check{R}_l-m-\check{r}-1}) \cdot C^b_m - (\check{r}+1) \cdot (\hat{S}^c_{\check{R}_l-\check{r}} + S^c_{\check{R}_l-\check{r}-1}) \cdot S^b_1 \\
&=\; \frac{\check{r}+1}{\check{r}}(A(0,\check{r}) \cdot S^b_0 + \check{r} \cdot \sum_{m=1}^{\check{R}_l-1} \hat{S}^c_{\check{R}_l-m-\check{r}} \cdot C^b_m - \check{r} \cdot \hat{S}^c_{\check{R}_l-\check{r}} \cdot S^b_1) + \\
&+(\check{r}+1) \cdot (\frac{\check{S}^c_{\check{R}_l-\check{r}} \cdot S^b_0}{\check{R}_l} + \sum_{m=1}^{\check{R}_l-1} S^c_{\check{R}_l-m-\check{r}-1} \cdot C^b_m - S^c_{\check{R}_l-\check{r}-1} \cdot S^b_1) \\
&=\; (\check{r}+1) \cdot (\frac{F(\check{r})}{\check{r}} + \frac{\check{S}^c_{\check{R}_l-\check{r}} \cdot S^b_0}{\check{R}_l} + \sum_{m=1}^{\check{R}_l-1} S^c_{\check{R}_l-m-\check{r}-1} \cdot C^b_m - S^c_{\check{R}_l-\check{r}-1} \cdot S^b_1)
\end{aligned}$$

The final solution can be written as:

$$LP_i \;=\; p_{L_i} \cdot \check{F}(\check{r}_{L_i}) + p_{H_i} \cdot \check{F}(\check{r}_{H_i}) + C^b_{over}$$

where

$$\check{F}(0) = 0 \quad ,$$

$$\check{F}(1) = \frac{S_0^b}{\check{R}_l} \cdot \left(\sum_{i=\check{R}_l-1}^{4\cdot\check{R}_l} i \cdot C_i^c - (\check{R}_l - 1) \cdot \check{S}_{\check{R}_l-1}^c \right) + \sum_{m=1}^{\check{R}_l-1} \hat{S}_{\check{R}_l-m-1}^c \cdot C_m^b - \hat{S}_{\check{R}_l-1}^c \cdot S_1^b \quad ,$$

and

$$\check{F}(\check{x}) = \check{F}(\check{x}-1) + \frac{\check{S}_{\check{R}_l-\check{x}+1}^c \cdot S_0^b}{\check{R}_l} + \sum_{m=1}^{\check{R}_l-1} S_{\check{R}_l-m-\check{x}}^c \cdot C_m^b - S_{\check{R}_l-\check{x}}^c \cdot S_1^b \quad ,$$

for $\check{x} \geq 1$.

$$S_j^c = \sum_{i=j}^{4\cdot\check{R}_l} C_i^c/i = S_{j+1}^c + C_j^c/j$$

$$\check{S}_j^c = \sum_{i=j}^{4\cdot\check{R}_l} C_i^c = S_{j+1}^c + C_j^c$$

$$\hat{S}_j^c = \sum_{i=j}^{\check{R}_l-1} S_i^c = \hat{S}_{j+1}^c + S_j^c$$

$$S_j^b = \sum_{i=j}^{\check{R}_l-1} C_i^b = S_{j+1}^b + C_j^b$$

$$C_{over}^b = \sum_{i=\check{R}_l}^{4\cdot\check{R}_l} C_i^b = 1 - S_0^b$$

Computational Complexity

$PL \rightarrow (2add + 2mul) \cdot n$

$\check{F}(\check{r} > 1) \rightarrow (3add + 3mul + (1add + 1mul) \cdot (\check{R}_l - 1)) \cdot (\check{R}_l - 1)$

$\check{F}(1) \rightarrow 4add + 4mul + 3 \cdot \check{R}_l \cdot (1add + 1mul) + (\check{R}_l - 1) \cdot (1add + 1mul)$

$S_j^c \rightarrow 4 \cdot \check{R}_l \cdot (1add + 1div)$

$\check{S}_j^c \rightarrow 4 \cdot \check{R}_l \; add$

$S_j^b \rightarrow \check{R}_l \; add$

$\hat{S}_j^c \rightarrow \check{R}_l \; add$

$C_j^c \rightarrow 4 \cdot \check{R}_l \cdot (2add + 3mul)$

$C_{over}^b \rightarrow 1add$

Taking into account that each of the divisions takes four times the time an addition or a multiplication requires for computation, the total number of calculations that must be performed in order to compute the loss probability for all of the n sources is given by:

$$4 \cdot n + 2 \cdot (\check{R}_l - 1)^2 + 52 \cdot \check{R}_l + 8 \cdot (\check{R}_l - 1) + 9$$

References

[1] M.Bonnati, A.Bovopoulos, A.Dailianas, A.Gaivaronski "Exact and Approximate Solution of a Multiplexing Problem," Technical Report WUCS-92-09, Department of Computer Science, Washington University in St. Louis.

[2] P.Boyer, D.Tranchier "A Reservation Principle with Applications to the ATM Traffic Control," Computer Networks and ISDN Systems 24, 1992.

[3] J.P.Coudreuse,P.Boyer,F.Guillemin "Spacing Cells for Peak Cell Rate Policing Purposes in ATM Networks," IEEE Network Magazine, Sept. 1992.

[4] J.R.Cox, J.S.Turner "Project Zeus: Design of a Broadband Network and its Application on a University Campus,"Technical Report WUCS 91-45, Department of Computer Science, Washington University in St. Louis, .

[5] A.I.Elwalid, D.Mitra "Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic," IEEE Networks Magazine, Sept. 1992.

[6] H.Esaki "Call Admission Control Method in ATM Networks," Proceedings of ICC 1992, pp. 1628-1633.

[7] S.J.Golestani, "Congestion-Free Communication in High-Speed Packet Networks," IEEE Transactions on Communications, Vol.39, No.12, Dec. 1991

[8] H.Heffes and D.M.Lucantoni "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," IEEE JSAC, Sept. 1986.

[9] J.Hennessy and D.Patterson, "Computer Architecture - A Quantitative Approach," Morgan Kaufmann Publishers, 1990.

[10] B.Maglaris, D.Anastassiou, P.Sen, G.Karlsson, J.Robbins "Performance Models of Statistical Multiplexing in Packet Video Communications," IEEE Transactions on Communications, Vol. 36, July 1988.

[11] Mahdavian, S. and Bovopoulos, A. D., "Effective Loss of Multiplexed ATM Cell Streams," Technical Report WUCS-93-08, Department of Computer Science, Washington University, 1993.

[12] G.Bianchi, V.Trecordi "Proposal for a comprehensive bandwidth management scheme and connection acceptance rule for B-ISDN," Technical Report WUCS-92-24, Department of Computer Science, Washington University, 1992.

[13] K.Sriram, W.Whitt "Characterizing Superposition Arrival Processes in Packet Multiplexors for Voice and Data," IEEE JSAC, Sept. 1986.

[14] J.S.Turner, "New Directions in Communications (or Which Way to the Information Age?)," IEEE Communications Magazine, v.24, No.10, Oct.1986.

[15] J.Turner, "Bandwidth Management in ATM Networks Using Fast Buffer Reservation," IEEE Networks Magazine, Sept. 1992.

[16] M.Schwartz, "Telecommunication Networks. Protocols, Modeling and Analysis," Addison-Wesley, 1987.

[17] D.Verma, D.Ferrari "Variation of Traffic Parameters in ATM Networks," Proceedings of ICC 1992.

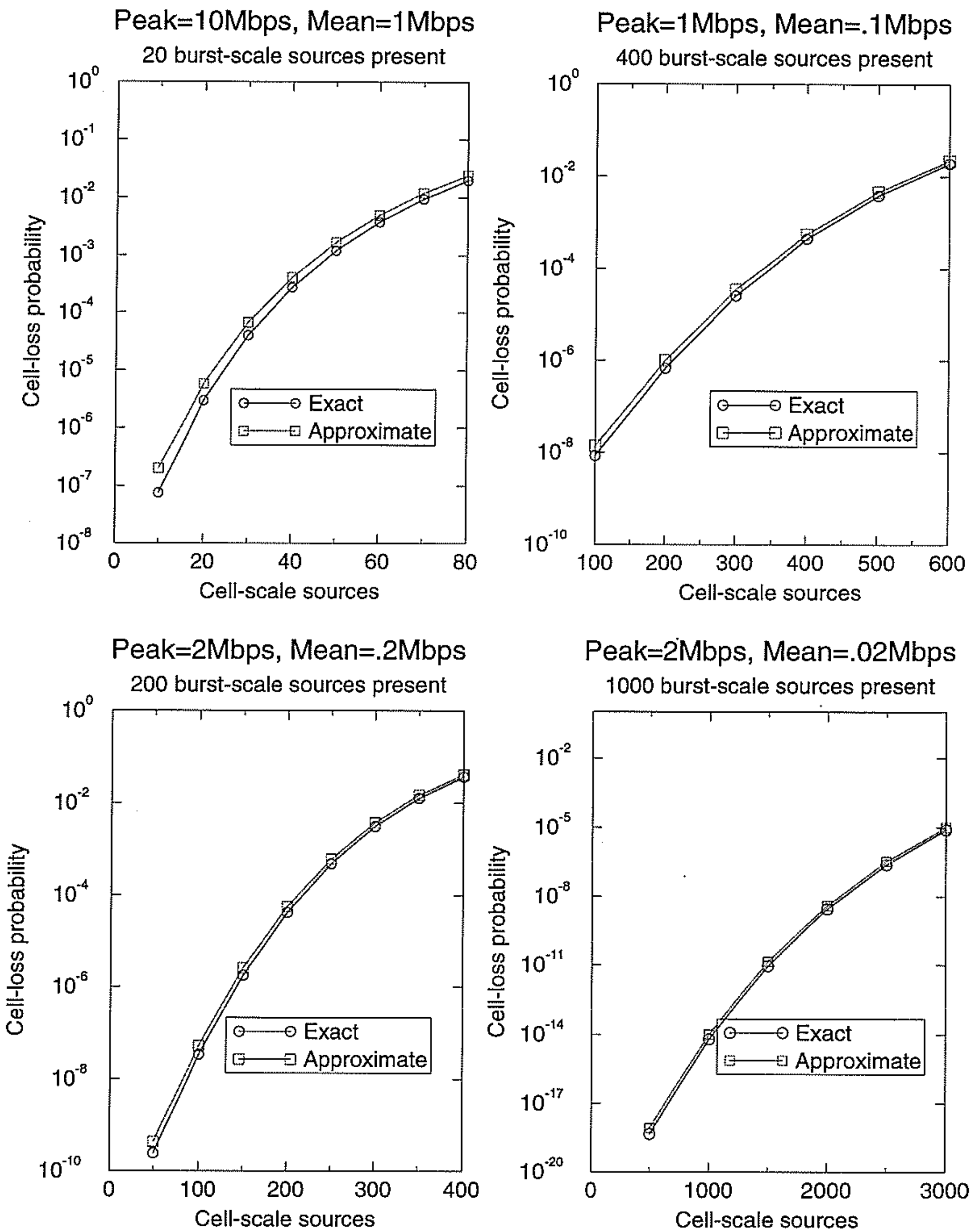


Figure 3: Effects of the approximation

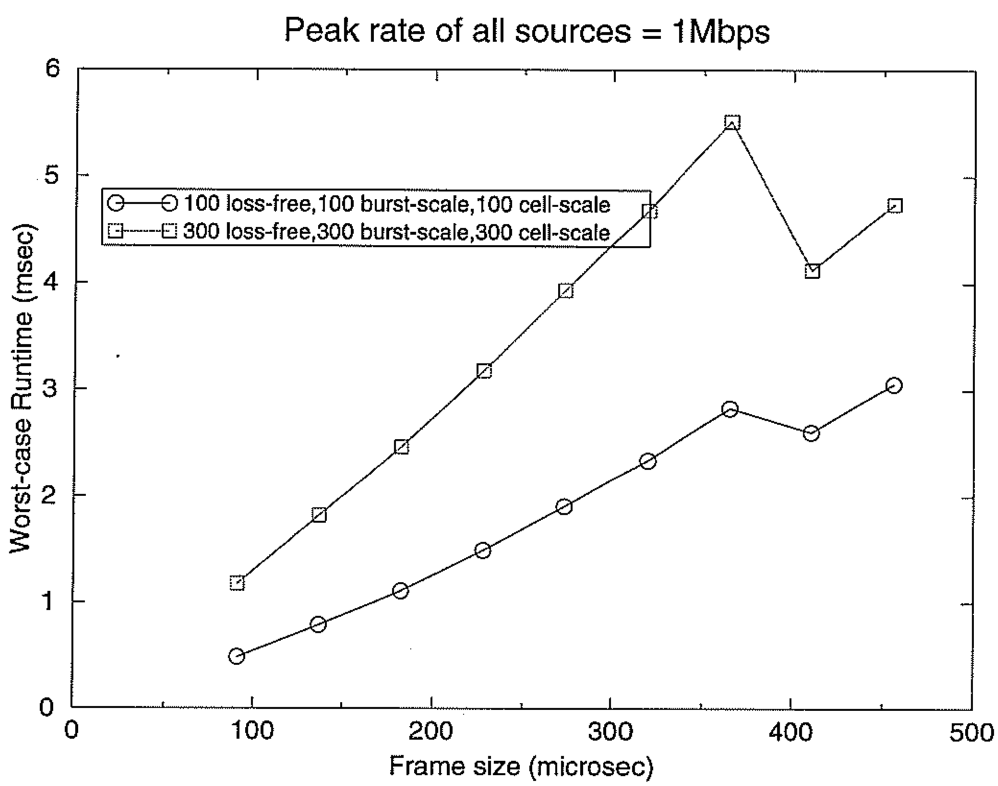
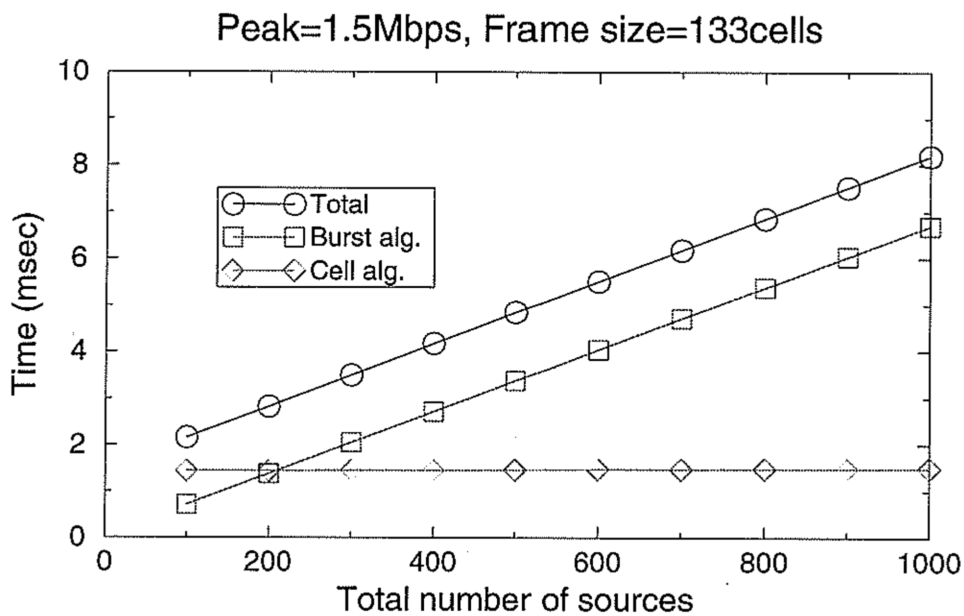


Figure 4: Time required for connection establishment for various frame sizes



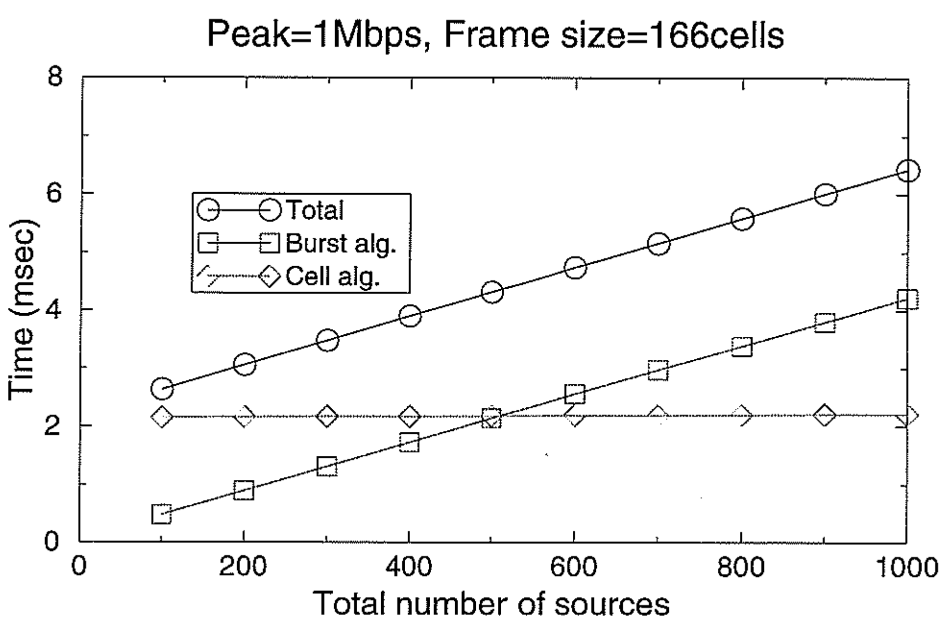


Figure 5: Time required for connection establishment as the number of sources increases

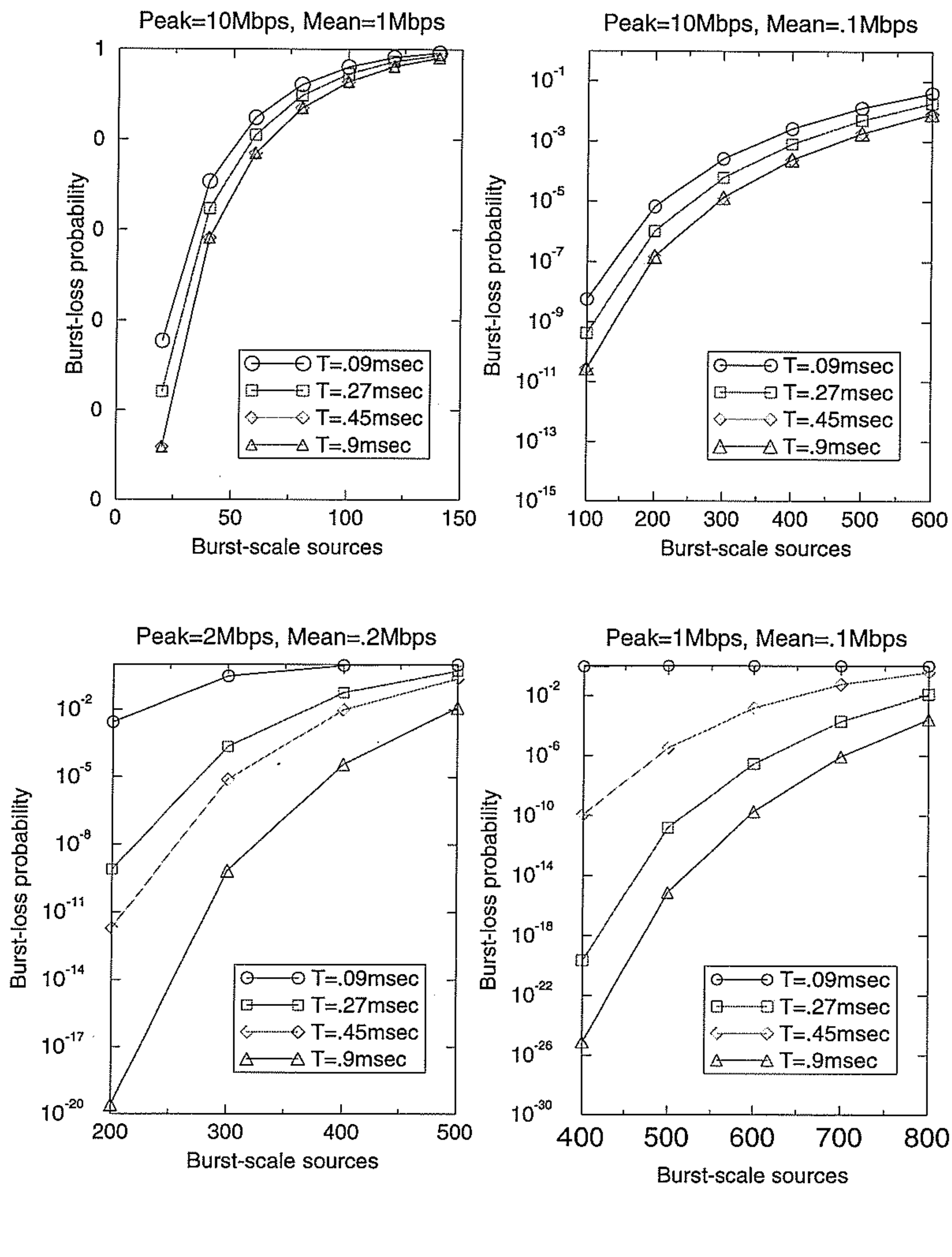


Figure 6: Frame size effect on burst-scale sources

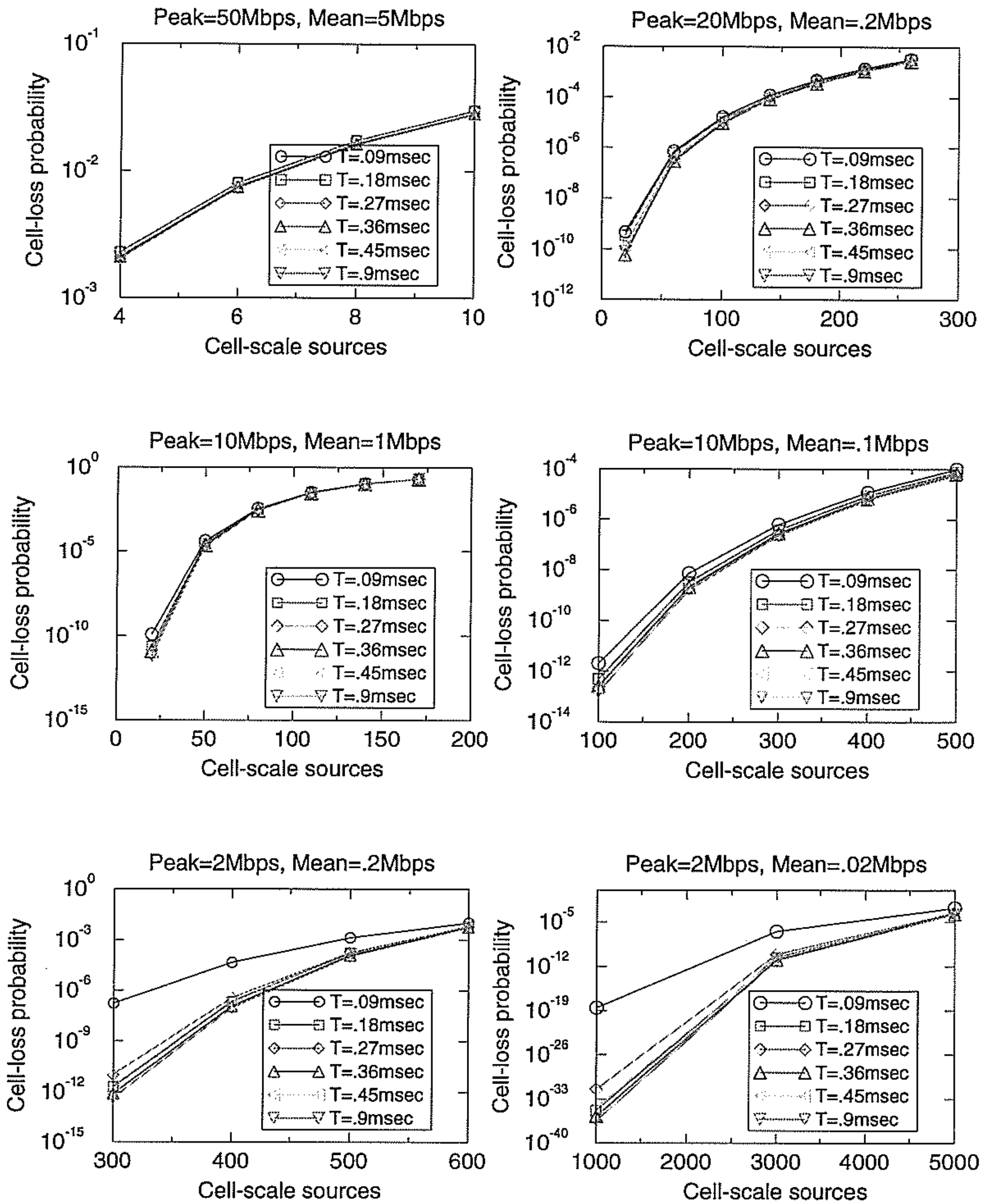


Figure 7: Frame size effect on cell-scale sources

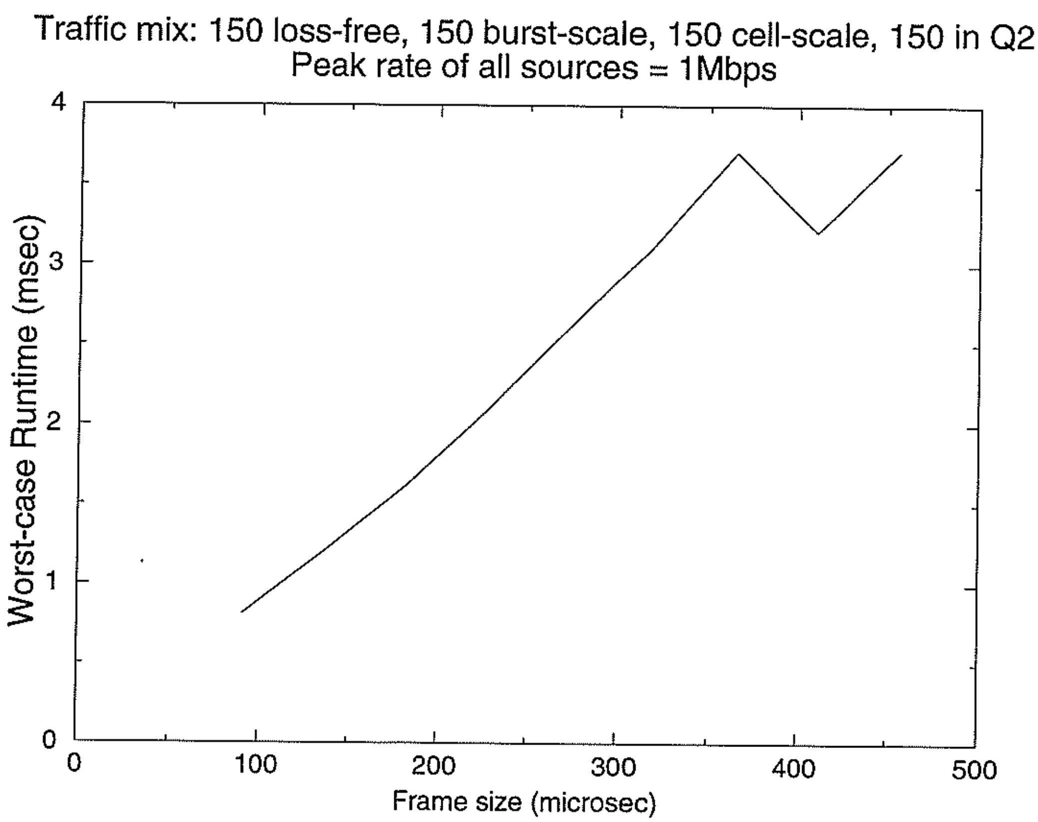


Figure 8: Time requirements including Q_2