

Adaptive Connection Admission Control for Mission Critical Real-Time Communication Networks

B. Devalla, A. Sahoo, Y. Guan, C. Li, R. Bettati and W. Zhao¹

Department of Computer Science, Texas A&M University, College Station, TX 77843-3112
{badari, asahoo, yguan, cenli, bettati, zhao}@cs.tamu.edu

{Paper submitted to MilCom '98}

Abstract

In this paper, we report our work on adaptive connection admission control in real-time communication networks. Much of the existing work on connection admission control (CAC) specifies the QoS parameters as fixed values and does not exploit the dynamic fluctuations in resource availability. We take an innovative approach: First, we allow an application to specify QoS in a range, rather than fixed values. Second and more importantly, we design, analyze, and implement CAC modules that, based on QoS specified over a range, adaptively allocate system resources to connections. Delay analysis is an integral part of connection admission control. Our adaptive CAC uses an efficient delay analysis method to derive a closed form solution for end-to-end delay of messages in a connection. With our adaptive CAC, we demonstrate an improvement in the system performance in terms of the probability of admitting connections and the QoS offered to the payload connections.

1 Introduction

In this paper, we report our work on an adaptive approach to connection admission control for providing effective and efficient connection admission control for mission critical applications distributed over real-time communication networks. These applications typically consist of a set of tasks executing on different hosts, exchanging messages over a high-speed network to co-operatively accomplish a common mission critical objective. Examples of such applications include supervisory command and control of defense systems, manufacturing plants, etc. The success of a distributed mission critical application thus crucially depends on the ability of the underlying network to guarantee upper bounds on message transfer delay.

Connection-oriented communication is well-suited for applications that demand performance guarantees [DLSZ97]. A connection can be viewed as a contract between an application and the connection management system. A *real-time* connection is additionally characterized by stringent deadline constraints imposed on its packet delivery time. The defining characteristic of connection-oriented communication is the existence of a connection

¹ Please send all correspondence to Prof. Wei Zhao, Dept. of Computer Science, Texas A&M University, College Station, TX 77843-3112. Tel: 409-845-5098, FAX: 409-847-8578, Email: zhao@cs.tamu.edu

establishment phase preceding the actual data transfer. Connection management is a network function that is responsible for setting up, maintaining and tearing down connections. A critical part of connection management is Connection Admission Control (CAC). The CAC determines whether a connection request can be admitted or not. Our study focuses on enhancing communication support for distributed mission critical applications through innovative connection admission control. With our adaptive strategy, we can demonstrate dramatic improvements in both the offered Quality of Service (QoS) to the applications, and the effective utilization of system resources.

Much work has been done on traditional connection admission control, which generally requires the QoS parameters to be specified as fixed values (e.g., traffic with peak 10 MBPS, and deadline 30 milliseconds). Once a connection is admitted, the traditional CAC provides a constant QoS to the connection throughout its lifetime. Thus, the traditional approach uses a simplistic QoS specification model and the consequent resource management suffers from many shortcomings that directly affect the applications using it. Specifically, this model is restrictive in that a fixed QoS model is not suitable for many applications that may accept admission at a lower QoS. For example, a video-on-demand application may be willing to accept a lower QoS (in terms of lesser bandwidth, jitter, etc.) to send video frames of poorer quality rather than send no frame at all. This model is also static, as the QoS offered to a connection does not change over its lifetime even though the resource availability changes. Consequently, the traditional CAC is very ineffective in terms of the number of connection requests admitted, as it neither exploits the dynamism of the system nor the flexibility in QoS suitable to applications. This also leads to a gross under-utilization of system resources.

To address the deficiencies of the traditional method, we take an adaptive approach. First, we allow an application to specify QoS in a range, rather than fixed values. Second and more importantly, we incorporate QoS adaptation that offers the best possible QoS to connections contingent on the available resources. Our approach has many benefits albeit offering deterministic performance guarantees on end-to-end message transfer delay. The probability of a connection being admitted is increased as the network has a choice over the range of QoS to offer. An adaptive resource allocation cognizant of the dynamic fluctuations in resource availability leads to a better utilization of system resources. In addition, at any given time, the existing connections are offered the best possible QoS allowed by the resources available.

This adaptive connection management is, nevertheless, a challenging proposition. The most important issue in the design is efficiency- defined in terms of how quickly an admission

decision can be made. An efficient CAC reduces the time between a connection request and the admission decision. Analysis of end-to-end delay plays an important part in determining the time taken for connection admission. Anytime a new connection request comes in or when the QoS of existing connections is changed, the adaptive CAC has to recompute the delays of existing connections. The adaptation mechanisms not only increase the complexity of delay analysis in an adaptive CAC but also invoke delay computation much more frequently than the traditional CAC. An adaptive CAC, therefore, needs an efficient delay analysis method. We address this issue by using a simple yet accurate traffic model and derive a closed form solution for end-to-end delay.

The rest of this paper is organized as follows: In Section 2 we present a brief survey of previous work in connection management. Section 3 presents an overview of adaptive connection management. Analysis of end-to-end message transfer delay is a key component of our approach. The delay analysis is presented in Section 4. An implementation and observations of our system serving distributed real-time applications is presented in Section 5. Section 6 concludes this paper with a summary of main results and suggestions for future extensions to this work.

2 Previous Work

The U.S. DoD has laid special emphasis on improving the responsiveness, security and reliability of communication services that play a critical role in current and future military operations [DISA94, ABIS96]. In this context, connection admission control for guaranteed performance (delay, jitter, etc.) is a well-researched topic. The communication is based on a simplex fixed-route connection called real-time channel, variations of the type defined in [Fer90]. A real-time channel is essentially a virtual circuit with performance guarantees. Various connection admission control approaches for guaranteeing performance of real-time channels have been suggested in [Par92, AKRS94, KSF94, MZ95, Cru95, MIS96, Rah96, DLSZ97]. Connection admission control is addressed in terms of scheduling policies, traffic regulation methods, and analysis of delay and buffer constraints. The aforementioned studies deal with QoS specified as fixed-values. Dynamism in connection management is discussed in [PVZ93, PZF94]. More recently, an architectural framework for adaptive resource management is reported in [HWVC97, HTG97, XJZ98]. Our study appropriately supplements previous work in developing communications service for mission critical applications. We identify and solve the important issues in adaptive connection admission control.

Our adaptive CAC module has been implemented in the newer version of our real-time toolkit, NetEx, over an ATM network. NetEx is a library of communication primitives that enables user applications to participate in delay guaranteed communications [DLSZ97, SLDZ97]. Our implementation is compatible with several standards and recommendations on QoS framework such as [ATM95, IT96].

3 Overview of Adaptive Connection Management

In this section, we present an overview of our adaptive connection management scheme. Connection management includes connection admission, maintaining connections and connection tear-down. We first discuss connection QoS specification and classification. We then discuss the adaptation strategies for connection admission and termination. For a more comprehensive discussion, please refer to [SDBZ98].

3.1 Connections

This sub-section outlines connection QoS specification and classification.

3.1.1 Connection QoS Specification

The periodic model is traditionally used to specify the QoS of a connection. The parameters are specified as the ordered triplet, (C, P, D), where C is the message size in bits generated periodically every P seconds, and each message has a deadline D seconds. The traditional model however specifies parameters with fixed values. We extend this model to specify parameters over a range of minimum and maximum values. The j-th connection has its QoS_j given by,

$$QoS_j = [QoS_j^{\text{worst}}, QoS_j^{\text{best}}] \quad (1)$$

where

$$QoS_j^{\text{worst}} = (C_j^{\text{min}}, P_j^{\text{max}}, D_j^{\text{max}}) \quad (2)$$

and,

$$QoS_j^{\text{best}} = (C_j^{\text{max}}, P_j^{\text{min}}, D_j^{\text{min}}) \quad (3)$$

If admitted, the connection is offered an operating QoS, QoS_j^{op} such that,

$$QoS_j^{\text{worst}} \leq QoS_j^{\text{op}} \leq QoS_j^{\text{best}} \quad (4)$$

Figure 1 shows this feasible QoS region. The objective of the adaptive CAC is to offer the best QoS possible to an admitted connection based on resource availability i.e., the adaptive CAC keeps QoS_j^{op} as close to QoS_j^{best} as possible.

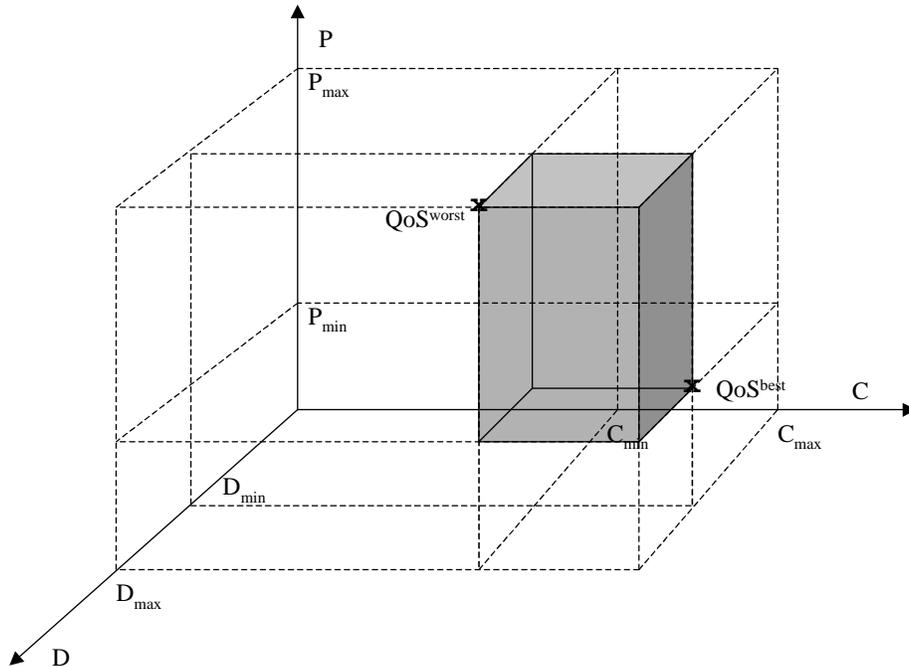


Figure 1. Feasible QoS Region

3.1.2 Connection Classification

Our scheme distinguishes connections to be from one of three classes: *critical*, *essential* and *non-essential*.

- *Critical* connections are of the highest criticality and are always admitted. Reserving resources a priori for critical connections ensures this. A critical connection, once admitted, cannot be preempted.
- *Essential* connections are of a criticality lower than critical connections but higher than the non-essential ones. An essential connection may be denied admission if sufficient resources are not available, but once admitted, it cannot be preempted.
- *Non-essential* connections are of the lowest criticality. They may be denied admission and be preempted in order to admit other connections of higher criticality.

Such a criticality-based classification is appropriate for mission critical system where tasks inherently are of different criticality [MZB90, HWC97, IT96]. In the absence of this classification, a less critical task would use the same amount of resources as a more critical one. This clearly results in poor management of resources especially when there is resource

contention. Further, classifying connections helps applications exploit the adaptation services during connection admission resulting in better overall performance.

3.2 Adaptation Strategies

We separate adaptation strategies in our adaptive connection management (See Figure 2) into two major threads - one for connection admission and another for connection termination. There is a *QoS Shrinkage* module for connection admission and a *QoS Expansion* module for connection termination.

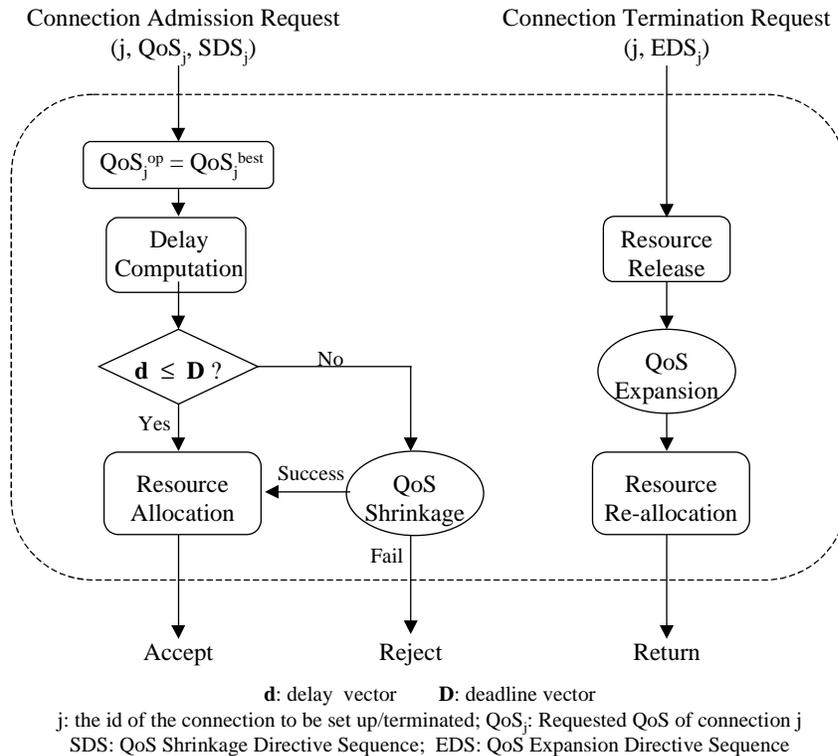


Figure 2. Adaptive Connection Management

3.2.1 Adaptive Connection Admission

A Connection Admission Request (CAR) comes with the following parameters: $\{j, QoS_j, SDS_j\}$, where QoS_j is the QoS of the j -th connection as defined in Equation (1). The connection request also specifies an adaptation policy via the Shrinkage Directive Sequence (SDS_j). SDS_j directs the connection admission module as to which connections' QoS need to be shrunk in order to admit the new connection.

For every incoming connection request, the QoS_j^{op} is set to QoS_j^{best} so that the new connection may be admitted at the best QoS requested if sufficient resources are available. The connection management scheme then computes the upper bound on the delay d due to the new

connection being admitted at QoS_j^{best} . Delay computation is a very important part of the connection admission process and is explained in detail in Section 4. We then test if all the requested deadlines can be met i.e. $\mathbf{d} \leq \mathbf{D}$.² The new connection is admitted only if the deadlines of this and the existing connections can be guaranteed.

If the guarantee test fails, the adaptation mechanism comes into play. Based on SDS, QoS Shrinkage determines the level to which the QoS of a selected set of connections need to be shrunk to successfully admit the new connection at QoS_j^{op} . If QoS Shrinkage is successful in admitting a connection by reducing QoS of some connections, resource allocation follows. A connection is rejected only when adaptation fails to free up enough resources to admit the new connection. Once a new connection is admitted, the adaptive CAC allocates resources for the new connection such that the connection operates at QoS_j^{op} .

The adaptation directive SDS allows user participation in the connection process. To admit a connection, the user specifies which connections' QoS is to be shrunk if necessary. This ensures adaptation by the CAC to be consistent with the user's mission objectives. There is also an added benefit to the efficiency of the adaptation scheme. The adaptive CAC is bereft of the complications of a selection process to determine which connections have to be subject to QoS shrinkage.

3.2.2 Adaptive Connection Termination

A Connection Termination Request comes with following parameters: $\{j, EDS_j\}$, where the adaptation policy of the j -th connection is specified by the Expansion Directive Sequence (EDS_j). EDS_j directs which connections are to get an increased QoS as result of resources released by the terminating connection.

When a valid CTR arrives, the adaptive connection management module releases all resources that were reserved for the connection during admission time. The adaptive connection management module based on EDS then determines the level to which the QoS of a selected set of connections can be increased using the resources released at connection termination. The necessary resources are then re-allocated to support the increased QoS.

² If there are N -connections, then \mathbf{d} is the vector of upper bounds of end-to-end delays (d_1, d_2, \dots, d_N) , and \mathbf{D} is the corresponding vector of deadlines (D_1, D_2, \dots, D_N) , $\mathbf{d} \leq \mathbf{D}$ iff $\forall i, d_i \leq D_i$, i.e. the worst-case end-to-end delay of every single connection is no greater than its respective deadline.

4 Delay Analysis

Delay analysis is an integral part of connection admission control. Whenever resources allocated to connections change, the delays of these connections have to be recomputed to ensure the new delays are less than their respective deadlines. Only then can the changes to resource allocation be committed. In case of adaptive CAC, allocated resources are changed quite frequently during the adaptation process. Hence delay computation is carried out much more often than in traditional CAC. Due to this frequent invocation, delay computation plays a pivotal role in the performance of adaptive CAC. Specifically, the efficiency of adaptive CAC largely depends on how quickly the delay can be calculated.

In this section, we will describe the delay analysis used in our adaptive CAC. The delay formula derived in this section is in a closed form. Thus it avoids search for the worst case value and contributes to the efficiency of the adaptive CAC. In this analysis, we will only concentrate on the network part of the delay. The network being analyzed consists of one ATM switch. The switch uses FCFS scheduling to schedule cells at the output port. Hosts are connected to each other via the ATM switch.

4.1 Traffic Description

Traffic description is very important for delay analysis. Accuracy and efficiency of delay analysis is very much dependent on information provided by the traffic descriptor. In mission critical systems, we are concerned with the worst case delays. Hence the traffic descriptor used, should represent the worst case traffic.

We had used the *maximum rate* function, $\Gamma(I)$ to represent traffic in some of our previous work [SLDZ97, RKZ95]. It is defined as the maximum data arrival rate in a time interval of length I . That is

$$\Gamma(I) = \max_{\forall t > 0} \left(\frac{\text{number of bits arrived in the interval } (t, t + I]}{I} \right) \quad (5)$$

The maximum rate function represents the worst case traffic quite accurately. It can be used to describe traffic at the intermediate points of the network. Analyzing a network with this method is much more accurate than when traffic is described only at the source. Furthermore, properties of maximum rate function are well established [Rah96]. Those properties are very helpful in the delay analysis.

But the maximum rate function has certain limitations. If a closed form is not available, then the value of the function at different values of time has to be stored, for the entire duration of a connection. This may not be practical in actual systems. Using maximum rate function, computation of worst case delay involves searching for the worst value in the set of all values. This search has a detrimental effect on the efficiency of adaptive CAC. Furthermore, when a connection passes through multiple servers, maximum rate function at the output of intermediate servers has to be calculated so that the next server can use it as input traffic. This adversely affects the efficiency of delay computation.

To avoid the limitations of maximum rate function, the *multi-point approximation* method was proposed in [FLRYZ96, SLDZ97]. In this method, traffic is specified at a certain number of points and traffic at unknown points is approximated using the information given at these known points. In this method, only the information at known points need to be stored, rather than storing information at every point (as in the case of maximum rate function method). Thus, the storage requirement is drastically reduced. Delay computation time is also reduced due to the fact that traffic only at known points are to be taken into consideration during delay analysis.

But the multi-point approximation method has also some drawbacks. As the number of known traffic points increases, the expression for calculating traffic at other points becomes very cumbersome [FLRYZ96]. Calculation of worst case delay still involves searching for the worst value in the set of all values. Typically, calculation of traffic at any point as a function of given points involves floor and ceiling functions in the expression. So the function becomes discontinuous and thus attempts to find worst case delay using delay calculus becomes intractable.

Hence, in this paper we propose *multi-segment approximation* of maximum rate function to represent traffic. In this method, we have the knowledge of the traffic at some known points. Then traffic is represented with as many line segments as the number of known points, each corresponding to the given points. This representation has the following advantages:

- Storage requirement is very low compared to maximum rate function method and is same as the multi-point method.
- Because traffic is represented as multiple line segments, the delay analysis is more tractable using delay calculus.
- A closed form delay formulation is possible in this representation of traffic. Hence this method eliminates the searching required to get the worst case delay. This

drastically reduces the delay computation time, which in turn, increases the efficiency of CAC.

To illustrate the multi-segment approximation method, let us assume the input traffic at the application layer of the connection M_i is given by periodic traffic C_i and P_i according to the periodic model described in section 3.1.1. This is one point of the traffic description which is given by the application. We also measure the packet length (C_{pkt}) in bits and the minimum distance between two consecutive packets (P_{pkt}) in seconds, as the packets exit the host and enter the network. This gives us the second point. The third point is obtained by measuring the cell length (C_{cell}) and the minimum distance between two consecutive cells (P_{cell}) at the entrance of the network. Further, we assume

$$C_i \geq C_{pkt} \geq C_{cell}$$

and

$$C_i/P_i \leq C_{pkt}/P_{pkt} \leq C_{cell}/P_{cell}.$$

(10)

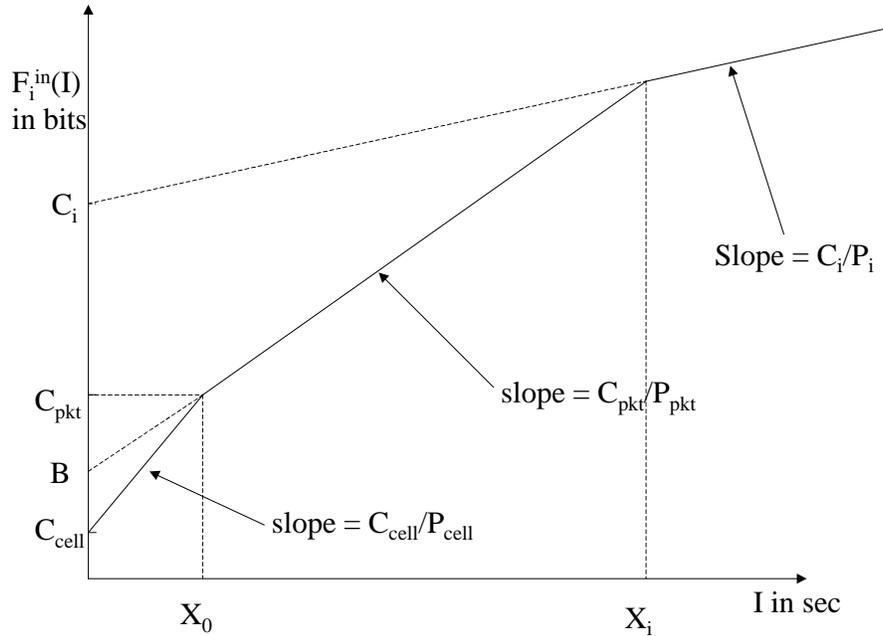


Figure 3: Plot of input traffic of a connection to the multiplexer

So the three points used to construct the three-segment approximation of traffic description for connection M_i are (C_{cell}, P_{cell}) , (C_{pkt}, P_{pkt}) and (C_i, P_i) . The three-segment approximation is shown in Figure 3. The first line segment starts at $I=0$, the second at $I=X_0$ and the third at $I=X_i$. The slopes of the three line segments are (C_{cell}/P_{cell}) , (C_{pkt}/P_{pkt}) and (C_i/P_i) respectively. The y-

intercepts of those lines are C_{cell} , B and C_i respectively. Using simple coordinate geometry principles, it can be easily established that B , X_0 and X_i are given by

$$B = C_{\text{pkt}} - C_{\text{pkt}}(C_{\text{pkt}}/C_{\text{cell}}-1)P_{\text{cell}}/P_{\text{pkt}}, \quad (7)$$

$$X_0 = (C_{\text{pkt}}/C_{\text{cell}}-1)P_{\text{cell}}, \quad (8)$$

and

$$X_i = [(C_i - C_{\text{pkt}}) + (P_{\text{cell}}/P_{\text{pkt}})C_{\text{pkt}}(C_{\text{pkt}}/P_{\text{pkt}}-1)] / (C_{\text{pkt}}/P_{\text{pkt}} - C_i/P_i) \quad (9)$$

So the three-segment approximation of input traffic to the multiplexer for connection M_i is given by

$$F_i^{\text{in}}(I) = \begin{cases} C_{\text{cell}} + R_{\text{cell}} \cdot I & 0 \leq I < X_0 \\ B + R_{\text{pkt}} \cdot I & X_0 \leq I < X_i \\ C_i + R_i \cdot I & X_i \leq I < \infty \end{cases} \quad (10)$$

where $R_{\text{cell}} = C_{\text{cell}}/P_{\text{cell}}$, $R_{\text{pkt}} = C_{\text{pkt}}/P_{\text{pkt}}$ and $R_i = C_i/P_i$.

4.2 Formulation of Delay

Network delay is the sum of the delay of a cell in the ATM switch (d_{atm}) and the delay of the cell to propagate from sender host to the switch and from switch to the receiver host (d_{prop}). Thus

$$d_{\text{network}} = d_{\text{atm}} + d_{\text{prop}} \quad (11)$$

d_{prop} is a constant delay and can be measured. ATM switch can be decomposed into the input port demultiplexer, the switching fabric and the output port multiplexer [Amitava DCS]. So delay in the ATM switch (d_{atm}) can be expressed as the sum of the delays at these three decomposed components.

$$d_{\text{atm}} = d_{\text{demux}} + d_{\text{fabric}} + d_{\text{mux}}, \quad (12)$$

where d_{demux} is the delay in the demultiplexer, d_{fabric} is the delay in the switching fabric and d_{mux} is the delay in the multiplexer. d_{demux} and d_{fabric} are constant delays and can be measured. But d_{mux} is a variable delay since it depends on the queue length at the output port. In our delay analysis, we will focus on calculation of d_{mux} . We use the three-segment approximation method to describe input traffic to the multiplexer.

Let us assume that there are N connections M_1, M_2, \dots, M_N that are passing through the multiplexer being analyzed. The input traffic at the application layer of connection M_i is periodic

traffic given by (C_i, P_i) . The three-segment approximation of input traffic of connection M_i is given by (10).

Since the first two points of all the connections are the same, the first bending point or flex point given by X_0 is same for all the connections. The second flex point X_i will be, in general, different for different connections. The total traffic $F^{in}(I)$ to the multiplexer is the summation of these individual $F_i^{in}(I)$ for all i , $1 \leq i \leq N$. So $F^{in}(I)$ will have at most $(N+1)$ flex points. Without loss of generality, we assume that flex points X_i , $0 \leq i \leq N$, of $F^{in}(I)$ are in a non-decreasing order i.e.,

$$X_0 \leq X_1 \leq X_2 \dots \leq X_N. \quad (13)$$

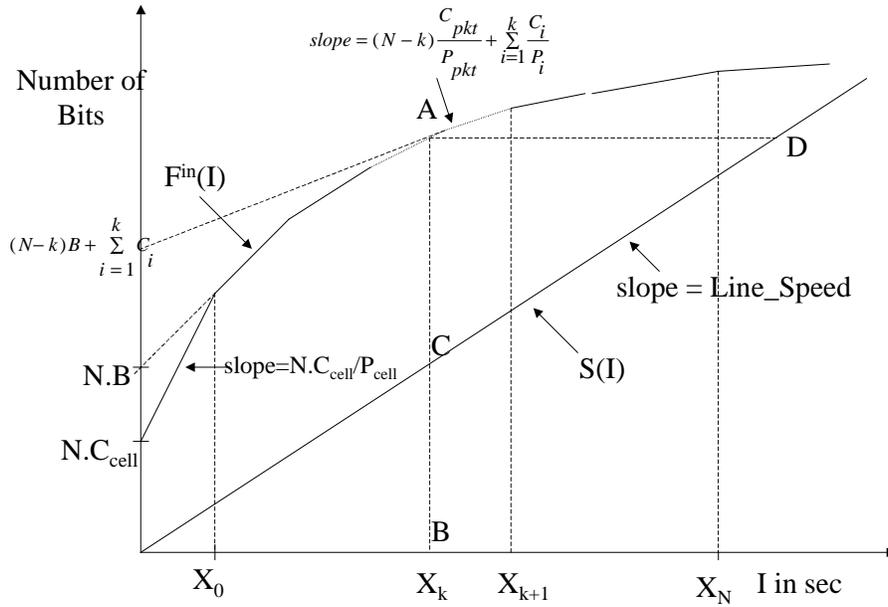


Figure 4 : Plot of the total traffic to a multiplexer

Then $F^{in}(I)$ is given by

$$F^{in}(I) = \sum_{i=1}^N F_i^{in}(I) = \begin{cases} N \cdot R_{cell} \cdot I + N \cdot C_{cell} & 0 \leq I < X_0 \\ N \cdot R_{pkt} \cdot I + N \cdot B & X_0 \leq I < X_1 \\ [(N-1) \cdot R_{pkt} + \sum_{i=1}^1 R_i] \cdot I + (N-1) \cdot B + \sum_{i=1}^1 C_i & X_1 \leq I < X_2 \\ \vdots & \vdots \\ [(N-k) \cdot R_{pkt} + \sum_{i=1}^k R_i] \cdot I + (N-k) \cdot B + \sum_{i=1}^k C_i & X_k \leq I < X_{k+1} \\ \vdots & \vdots \\ \sum_{i=1}^N R_i \cdot I + \sum_{i=1}^N C_i & X_N \leq I < \infty \end{cases} \quad (14)$$

The plot of $F^{in}(I)$ is shown in Figure 4.

Also shown in Figure 4 is the service curve $S(I)$ of the multiplexer. The multiplexer serves at a constant rate of $Line_Speed$ (in bits/second) whenever there are cells to be served. $Line_Speed$ is 155 Mb/s for ATM switches with OC3 ports. Hence $S(I)$ is a straight line with slope equal to $Line_Speed$ passing through the origin.

Once the total traffic to the multiplexer and the service function of the multiplexer are determined, basic delay calculus [Cru95] can be applied to find the worst case delay. For any I , $F^{in}(I)$ represents the number of bits that has arrived at the multiplexer and $S(I)$ shows the number of bits served by the FCFS server at the multiplexer. Hence the difference $F^{in}(I) - S(I)$ shows the worst case queue length at the multiplexer during any time interval of length I . Then the worst case delay at the multiplexer occurs when this difference $F^{in}(I) - S(I)$ is the maximum. Since $F^{in}(I)$ consists of line segments, the difference will be maximum at one of the flex points where the input data rate (slope of $F^{in}(I)$) changes from more than $Line_Speed$ to less than $Line_Speed$. Let this particular flex point be X_k , $0 \leq k \leq N$. Then the worst case delay in the multiplexer is given by

$$d_{max} = \begin{cases} [(N-1) \cdot X_0 \cdot R_{cell} + N \cdot C_{cell}] / R_{cell} & k = 0 \\ [(N-k)B + \sum_{i=1}^k C_i - X_k \{Line_Speed - (N-k)R_{pkt} - \sum_{i=1}^k R_i\}] / Line_Speed & 0 < k \leq N \end{cases} \quad (15)$$

In Figure 4, the worst case queue length is represented by AC. The corresponding worst case delay is given by AD. We cannot provide the proof of (15) due to space limitation. Complete proof can be found in [Li98].

4.3 Usage of Delay Formula by Adaptive CAC

In this section we will show the steps used by adaptive CAC to recalculate delay when operating QoS of some of the connections is changed during the adaptation process. Recall that adaptation process starts when the system does not have enough resources to admit a new connection with its QoS^{best} . In that case, the CAC shrinks operating QoS of a set of connection according to the SDS given in the request. The two points (C_{cell}, P_{cell}) and (C_{pkt}, P_{pkt}) are obtained by measurement and is stored once and for all. The first flex point, X_0 is calculated using (8). The adaptive CAC recalculates delays as follows:

- Let us assume that the operating QoS of a set of connections G is reduced during adaptation to admit a new connection M_j . The set G is determined by the SDS given by the user in the request.
- Due to change in operating QoS, second flex point of connections in G will change. So the new second flex points X_i 's are calculated for all $M_i \in G$.
- Let $G_1 \supseteq G$ be the group representing connections that pass through the same output port as M_j . Let there be N connections in G_1 . Because the second flex point of connections in G changed, they are sorted in a non-decreasing order X_1, X_2, \dots, X_N .
- Using (10) new traffic description $F_i^{\text{in}}(I)$ of each individual connections $M_i \in G$ is calculated.
- $F_i^{\text{in}}(I)$ for all $M_i \in G_1$ are added up to find the aggregate traffic $F^{\text{in}}(I)$ using (14).
- From $F^{\text{in}}(I)$, the flex point $X_k, 0 \leq k \leq N$, is found such that the rate of $F^{\text{in}}(I)$ changes from more than Line_Speed to less than Line_Speed . This point corresponds to the worst case delay.
- Using (15) d_{mux} is calculated. This delay is same for all the connections $M_i \in G_1$. This delay is then added up with other constant delays in (12) to get the delay in the switch.

Thus it is apparent from the steps that the delay calculation involves no costly search. The most costly operation in this delay calculation is sorting the X_i 's which can be done in a logarithmic complexity. The final delay expression given by (15) does not need adaptive CAC to search for the worst case delay, the delay is calculated by plugging the value of X_k . Thus three-segment approximation method of traffic description leads to an efficient delay computation.

5 Implementation and Evaluation

5.1 Implementation in NetEx

Our adaptive CAC module has been implemented in the newer version of our real-time toolkit, NetEx [SLDZ97]. NetEx is a communication software package developed in the Department of Computer Science at Texas A&M University. NetEx is a library of communication primitives that enables user applications to participate in delay guaranteed communications. NetEx consists of three main components: user library, Host Traffic Manager (HTM) and Network Traffic Manager (NTM). User library is the interface of NetEx to the end users. HTM is the module responsible for managing and policing traffic at the host. NTM is

primarily responsible for connection management of the entire system. For a detailed description of NetEx architecture please refer to [SLDZ97].

5.2 Performance Metrics and Observations

We use the following performance metrics to compare the traditional and adaptive approaches to connection admission control:

- Admission Probability (AP): Admission Probability of a connection admission control system indicates the likelihood of a connection being admitted. We define admission probability as

$$AP = \frac{\text{Number of connections admitted}}{\text{Total number of connections requested for admission}}$$

- QoS Effectiveness (QoSE): The QoS effectiveness of a connection is a measure of how close the operating QoS of the connection is to the maximum QoS asked by the connection. It is defined as

$$QoSE = \frac{\sqrt{(C^{op} - C^{\min})^2 + (P^{op} - P^{\max})^2 + (D^{op} - D^{\max})^2}}{\sqrt{(C^{\max} - C^{\min})^2 + (P^{\min} - P^{\max})^2 + (D^{\min} - D^{\max})^2}}$$

- Average Execution Time (AET): Execution time is the amount of time an application has to wait to get a reply from the connection admission control module after submitting a connection admission request. Average Execution Time of N connections is defined as

$$AET = \frac{\text{Sum of execution time of } N \text{ connections}}{N}$$

Table I shows AP, QoSE and AET for the traditional and adaptive CAC for low, medium and high system utilization. System utilization is a measure of resources demanded by incoming connection requests.

Table I Performance of Traditional and Adaptive CAC Vs. System Utilization

Metric	AP (%)		QoSE		AET (milliseconds)	
	<i>Traditional</i>	<i>Adaptive</i>	<i>Traditional</i>	<i>Adaptive</i>	<i>Traditional</i>	<i>Adaptive</i>
Low	98.9	99.1	1.0	0.998	0.10	0.48
Medium	94.6	96.1	1.0	0.986	0.09	0.49
High	82.5	87.0	1.0	0.899	0.09	0.62

As system utilization increases, AP of both the traditional and adaptive CAC decreases. This is because the availability of system resources decreases as utilization increases and more connections are rejected. However the AP for the adaptive approach is always higher than that for the traditional scheme. The traditional scheme always accepts connections at QoS^{best} and hence allocates more resources to each admitted connection than the adaptive scheme. As a result, the AP of the traditional scheme is lower.

QoSE is high for the traditional CAC as it admits connections at QoS^{best} . For the adaptive CAC, QoSE is closer to traditional CAC at low and medium utilization. This is because at low and medium utilization, more resources are available, so connections can be admitted with operating QoS equal to or very close to QoS^{best} . But at high utilization, QoSE for the adaptive CAC drops as it tries to admit more connections at a lower QoS.

AET of both the traditional and adaptive schemes increases as system utilization increases. At higher utilization, both the systems have more active connections to deal with, which makes the execution time longer. AET of the adaptive CAC is more than traditional CAC as the adaptation expends time in shrinking and expanding the QoS while trying to admit and terminate connections. Nevertheless, the overall AET of the adaptive CAC is very low (less than 1 millisecond), which is very good for systems in practice.

6 Conclusion and Future Work

In this paper we have introduced adaptive connection admission control that addresses the shortcomings in traditional connection admission control. We have demonstrated that by taking an adaptive approach to connection admission control, we can enhance communication support for mission critical real-time applications. The highlights of this study are:

- *A Flexible QoS model:* We have extended the traditional fixed-value QoS model to one that accepts QoS specified over a range. This gives the applications and the admission control flexibility in resource allocation.
- *Criticality-based connection model:* Our adaptive CAC distinguishes between connections based on criticality. This is particularly suitable to mission critical applications that need criticality-based connection admission.
- *Efficient and effective QoS adaptation modules:* With QoS Shrinkage and QoS Expansion modules, the resources are dynamically re-allocated in order to meet the needs of incoming connections. Our performance data shows that the additional cost (in terms of execution time) of adaptation is low while the benefits are high.

- *User-level adaptation directives:* These allow user participation in the connection admission control process. This further enhances connection-level flexibility, which is useful for mission critical applications.
- *Practical and compatible technology:* The proposed adaptive connection admission control scheme has been implemented in NetEx. NetEx is a toolkit of communication primitives for delay-guaranteed communications. Data collected from our experiments with the implementation confirms our thesis that the overall performance of the system is improved when connection admission control responds to dynamic fluctuations in resource availability. Our proposed scheme is compatible with existing network standards and industrial practices. NetEx is realized with network products that are currently commercially available and does not require any changes to them.

Several extensions to adaptive connection admission control are possible. One immediate extension is to support delay guarantees in multi-hop switched networks with hosts interconnected by different high-speed LAN technologies like FDDI and ATM. We are also currently adding fault-tolerant techniques so that the overall communication system will adaptively meet the real-time and fault-tolerance requirements of mission critical applications.

7 Reference

- [ABIS96] ABIS Task Force, DoD Advanced Battlespace Information Systems Task Force Report, 1996.
- [AKRS94] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne, "Real-time communication in packet-switched networks," *Proceedings of the IEEE*, vol. 82, pp. 122-139, Jan. 1994.
- [ATM95] ATM Forum, ATM Forum- ATM User-Network Interface Specification Version 3.1, 1995.
- [BKHDP96] G. v. Bochman, B. Kerherve, A. Hafid, P. Dini, A. Pons, "Architectural Design of Adaptive Distributed Multimedia Systems," *IEEE International Workshop on Multimedia Software Development*, 1996, pp. 31-40
- [Cru95] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal of Selected Areas in Communication*, special issue on "Advances in the Fundamentals of Networking" (vol. 13 no. 6 pp. 1048-1056), August 1995.
- [DISA94] Defense Information Systems Agency, Baseline Common Operating Environment, 1994.
- [DLSZ97] B. Devalla, C. Li, A. Sahoo, W. Zhao, "Connection-Oriented Real-Time Communication for Mission Critical Applications: An Introduction to NetEx," *Proceedings of NAECON '97*, pp. 698-707, July 1997.
- [Fer90] D. Ferrari, "Client requirements for real-time communication services," *IEEE Communications Magazine*, vol. 28, pp. 65-72, Nov. 1990.
- [FHBM97] S. Fischer, A. Hafid, G. Bochmann, H. De Meer, "Co-operative QoS Management for Multimedia Applications," *Proceedings of IEEE Intl. Conf. on Multimedia Computing and Systems 1997*, pp. 303-310.

- [FLRYZ96] F. Feng, C. Li, A. Raha, S. Yu, and W. Zhao, "Modeling and Regulation of Host Traffic in ATM Networks for Hard Real-Time Applications," *Proc. of 21st IEEE LCN*, 1996.
- [HTG97] J. Huang, R. Jha, W. Heimerdinger, M. Muhammad, S. Lauzac, B. Kannikeswaran, K. Schwan, W. Zhao, R. Bettati, "RT-ARM: A Real-Time Adaptive Resource Management System for Distributed Mission-Critical Applications," *IEEE Workshop on Middleware for Distributed Real-Time Systems and Services 1997*
- [HWC97] J. Huang, Y. Wang, F. Cao, "On Developing Distributed Middleware Services for QoS- and Criticality-Based Resource Negotiation and Adaptation," Honeywell Technical Report SST-P97-012, June 1997.
- [HWVC97] J. Huang, Y. Wang, N.R. Vaidyanathan, F. Cao, "GRMS: A Global Resource Management System for distributed QoS and Criticality Support," *Proceedings of COMPSAC'97*, pp. 573-579.
- [IT96] "Information Technology - Quality of Service Framework," DIS 13236, ISO/IEC JTC1/SC21 N10339, June 1996.
- [KSF94] D. D. Kandlur, K. G. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, pp. 1044-1056, Oct. 1994.
- [Li98] C. Li, "Design of Adaptive Connection Admission Control for Hard Real-time ATM Networks," M.S. Thesis, Texas A&M University, May 1998.
- [MIS96] A. Mehra, A. Indiresan, and K. Shin, "Resource management for real-time communication: Making theory meet practice." *Proceedings of RTAS*, 1996.
- [MZ95] N. Malcolm and W. Zhao, "Hard Real-Time Communication in Multiple- Access Networks," *Journal of Real-Time Systems*, January 1995.
- [MZB90] N. Malcolm, W. Zhao, C. Barter, "Guarantee Protocols for Communication in Distributed Hard Real-Time Systems," *Proc. of IEEE INFOCOM'90*, San Francisco, USA 1990, pp. 1078-1086.
- [Par92] A. K. J. Parekh, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1992.
- [PVZ93] C.J. Parris, G. Ventre, H. Zhang, "Graceful Adaptation of Guaranteed Performance Service Connections", *Proceedings of GlobeCom 1993*.
- [PZF94] C.J. Parris, H. Zhang, D. Ferrari, "A Dynamic Management Scheme for Real-Time Connections," *Proc. IEEE INFOCOM '94*, Toronto, Ontario, Canada 1994, pp. 698-707.
- [Rah96] A. Raha. *Real-Time Communication in ATM Networks*. Ph.D. Thesis. Texas A&M University, 1996.
- [RKZ95] A. Raha, S. Kamat, W. Zhao, "Guaranteeing End-to-End Deadlines in ATM Networks", *Proc. of International Conference on Distributed Computing System 1995*.
- [SDBZ98] A. Sahoo, B. Devalla, R. Bettati, W. Zhao, "Adaptive Connection Management in NetEx," Technical Report, Computer Science Department, Texas A&M University, 1998.
- [SLDZ97] A. Sahoo, C. Li, B Devalla, W. Zhao, "Design and Implementation of NetEx: A Toolkit for Delay Guaranteed Communications," *Proceedings of MilCom '97*.
- [XJZ98] Y. Xue, W. Jia, W. Zhao, "Efficient Connection Admission Control Algorithms for Adaptive Real-Time Connections over ATM LANs," submitted to *INFOCOM 98*.