Graceful Adaptation of Guaranteed Performance Service Connections*

Colin J. Parris, Giorgio Ventre† and Hui Zhang

The Tenet Group
Computer Science Division, Department of EECS
University of California, Berkeley
and
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704.

Abstract

Most of the solutions proposed to support guaranteed performance communication services in a packet-switching network adopt a connection-oriented and reservation-oriented approach. In this approach, the resource allocation and route selection decisions are made during connection establishment on the basis of the current resource availability and real-time[1] network load, and are usually kept for the duration of the connection's lifetime. However, such an approach shows two major limitations: first, the communication service provided is usually fixed, with limited or no capability of adapting to dynamic changes in the clients' requirements; second, a low utilization of the network may be observed. In this paper, we present a flexible management scheme that allows graceful adaptation of guaranteed performance service connections. Mechanisms have been devised to allow changing of the traffic and performance parameters of a real-time connection during its lifetime. These mechanisms, together with an adaptation policy, can make more efficient use of the network resources by performing cooperative, consenting, high-level multiplexing. We distinguish between two types of adaptation: client initiated adaptation and network initiated adaptation. We give examples for both types and we also present results from simulation experiments to show the effectiveness of our approach.

*This research was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing official policies, either expressed or implied, of the U.S. Government or any of the sponsoring organizations.

†Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II", Napoli, Italy.

[1] In our context we refer the guaranteed performance characteristics of a connection as real-time.

1 Introduction

High speed networking is introducing opportunities for new multimedia applications such as video conferencing, scientific visualization and medical imaging. These applications have stringent network performance requirements in terms of parameters such as bandwidth, delay, delay jitter, loss rate or some combination of these. Guaranteed performance service communication is needed to support these applications as the best-effort service provided by the traditional packet-switching networks is not adequate [4].

Several solutions have been proposed to support guaranteed performance service communication in packet-switching networks [1, 6, 11, 13]. They all adopt a connection-oriented and reservation-based approach. In such an approach, a connection is established within the network, and resources are reserved for it so that the performance requirements of the application are met. Usually, the resource allocation and route selection decisions are made at the establishment of the connection on the basis of resource availability and real-time traffic load, and are kept for the duration of the application lifetime. However, the network state and the amount of resources needed by an application may change during the life time of an application. For example, in a video conferencing application, there may be only two participants at the beginning, with more participants joining later. The video connections established at the beginning of the conference may have good quality; however, as more participants join the conference it may not be feasible to support all the participants at the same quality of service, due to resource constraints, as when there are only two participants. It may be more desirable to degrade the quality of service of the established connections rather than to deny new participants access to the video conference.

In this paper we present an approach that allows the adaptation of guaranteed performance connections based on the dynamicity of the network load and client requirements. This approach is based on the DCM algorithms which allow the modification of the traffic and performance parameters of a guaranteed performance connection during its lifetime. The parameter modification can be initiated either by the network or by the client. In *network* *initiated adaptation*, a client should specify, at the time of the request of a connection, a range instead of a single value for all or some of the parameters characterizing the quality of service of the connection. The network can then change the parameters of the connection within the specified range by adjusting the resources allocated to the client (with the client's consent), based on the current real-time load in the network. In *client initiated adaptation*, parameter changes are initiated by clients to reflect the dynamicity of its requirements. For example, in a scientific visualization application, the user may want to vary the speed of the visualizing process. This can be achieved by dynamically changing the parameters of the underlying network connection.

We believe that the introduction of these adaptation mechanisms are beneficial for both the client of real-time communication services and the network. For the former, these mechanisms should increase the probability of accepting the client's connection request and enhance the client's control over the allocation of the resources to its needs. For the latter, the high level *cooperative, consenting, high-level multiplexing* introduced by these mechanisms should improve the overall network utilization.

There are two major assumptions at the basis of this approach. The first is that a number of applications can accept a varying quality of service if the dynamics of the quality of service is under the client's control and is within well specified ranges. The second is that these applications will appreciate this flexibility as it provides a higher probability of having the connection request accepted by the network [3], and possibly also a pricing scheme advantageous to the client [8].

Our scheme of dynamically managing connection parameters is within the framework of a guaranteed performance communication service. All the changes are either initiated by the client or pre-granted. In either case, the network is still committed to guaranteeing a service strictly within the performance ranges specified by the client. Another important feature of our proposal, that makes it particularly suited for its application in real-time communication networks with guaranteed service, is that it is able to provide a *graceful adaptation of service*, i.e. a change from the old quality of service to the new quality of service with limited or no disruption.

The paper is organized as follows: in Section 2, related work is reviewed. The mechanisms comprising the adaptation schemes are presented in Section 3. Two mechanisms are discussed; the first one is network initiated while the second is initiated by the clients. These mechanisms are all based on the Dynamic Connection Management (DCM) [9] algorithms, which are extensions of the Tenet [6] algorithms. A simple adaptation policy is also presented in Section 3. Simulations and their analysis are provided in Section 4, and the paper concludes in Section 5 .

2 Related Work

The issue of adaptation of guaranteed performance communication services has only been partially addressed in the literature.

In the ST-2 protocol [11], the ideas of a flow of information from the network back to the client, and of dynamic modification of the performance characteristics of a connection are introduced. However, since the protocol specification does not present any scheme for resource management that could provide the service guarantee, it is not clear how solutions to the two problems mentioned above can be implemented.

Recently, two different solutions have been proposed to increase the adaptivity of real-time networks. In [2], a new service, called *predicted*, is introduced in which the performance delivered by the network in terms of delay bounds is computed by measuring the current load, rather than by using precomputed, worst-case data. This service is proposed for a particular class of multimedia applications, called *playback applications*. In these applications, packets received at the destination are buffered, to remove the network induced jitter, to be successively depacketized and played at some playback point designated using the network's measurements. Since in predicted service, fluctuations in the network load can produce variations in the provided quality of service, the network cannot commit to offer a well determined performance. In addition, applications are required to be capable of enduring a high packet loss rate and even service disruption due to the adaptation of the playback point to the variations of the end-to-end communication delay.

In the extended version of the CBSRP protocol [10], the user can specify the minimum and maximum values (i.e. the minimum and maximum quality of service) for two parameters: the desired temporal and spatial resolutions of the media to be transmitted. The specified values allow the network to assign each client to a particular class of service. When a new client requires the establishment of a session, if the available resources are already saturated, some existing sessions may be forced to reduce their quality of service, to accommodate the new request. The minimum quality of a session is, however, always guaranteed once the session is established. The limitation of CBSRP is that the quality of service is specified only in terms of inter-packet distance and packet size, while delay, delay jitter, and packet losses are not taken into account.

In the context of supporting video service in datagram networks, algorithms have been proposed to adapt coding parameters and vary the output rate using the feedback signal from the network [7, 12]. Their proposals differ from ours in that they assume a connectionless network and thus no performance guarantees are provided.

3 Graceful Adaptation

In this section, we present the mechanisms that provide the channels[2] with the ability to modify their performance parameters. This mechanism is the Dynamic Connection Management (DCM) scheme which is an extension of the Tenet Scheme. Initially an overview of the Tenet Scheme will be presented, followed by an overview of the DCM scheme. The two adaptation services, network initiated and client initiated, will then be discussed. A simple adaptation policy will also be presented in the final subsection.

[2]There are different terms in the literature for the same or similar objects. In this paper we refer to a guaranteed performance connection as a channel.

3.1 The Tenet Model

In the Tenet Scheme, a guaranteed performance service connection (*a.k.a. a real-time channel*) is a communication abstraction that defines real-time communication services associated with traffic and performance parameters in a packet-switched network in accordance with the Tenet model [5].

A channel's real-time traffic is characterized by the following parameters: X_{min}, the minimum packet inter-arrival time, X_{ave}, the average packet inter-arrival time over an averaging interval I, and , S_{max}, the maximum packet size. The performance requirements available to a channel are: D, the maximum delay permissible from the source to the destination, J, the maximum delay jitter,[3] Z, the probability that the delay of the packet is smaller than the delay bound, D, and, W, the buffer overflow probability.

A channel is *established* before data transfer. This channel establishment is achieved, in the following manner: a real-time client specifies its traffic characteristics and end-to-end performance requirements to the network; the network determines the most suitable route for a channel with these traffic characteristics and performance requirements; it then translates the end-to-end parameters into local parameters at each node, and attempts to reserves resources at these nodes accordingly. If the needed resources can be reserved at the nodes, the channel is accepted otherwise it is rejected.

The Tenet algorithms are used to determine whether a node has sufficient resources to accommodate a channel request. These algorithms or admissions tests are based on the service discipline in the nodes and the traffic model used. After the channel has been established, the data transfer phase commences. By appropriate scheduling and rate control, the local performance requirements are met at each node, and the client specified end-to-end performance guarantees are thus satisfied.

3.2 The DCM Scheme

The motivation for Dynamic Connection Management (DCM) was to increase the flexibility and availability of real-time network services. To this end, a collection of algorithms were developed to enable the network to dynamically modify the traffic characteristics and the performance requirements of a real-time channel, and to modify the route traversed by the channel. These modifications can also be performed in a manner that is transparent to the client. A complete description of the DCM scheme is given in [9].

The modification of a real-time channel is a procedural abstraction whereby a real-time channel with the new performance parameters (referred to as the alternate channel) is established, the client's traffic is moved from the current real time channel (referred to as the primary channel) to the alternate channel, and then the primary channel is removed. The movement of traffic from the primary to the alternate channel is referred to as the *transition* from the primary to the alternate channel. Using a DCM Modification contract, this transition can be accomplished with no performance violations or a bounded number of performance violations. The performance guarantees that can be violated are the guarantees on the throughput bound, delay bound, delay-jitter bound, and packet ordering. The DCM Modification Contracts are contractual obligations made to the client that determine the extent of the performance violations that can occur during the transition interval. There are two types of contracts; in the first type no performance violations will occur, while, in the second, a bounded number of performance violations can happen during a bounded interval of time called the *transition interval*.

The DCM scheme consists of three algorithms: the *Channel Administration* algorithm, the *Transition* algorithm, and the *Routing* algorithm. The *Channel Administration* algorithm determines whether or not a real-time channel can be accepted along a specified alternate route and, if so, reserves the resources along the alternate route so that all of the client's traffic and performance requirements are met. The algorithm is also responsible for recovering the resources that were previously allocated to the primary channel after the transition from the primary to the alternate channel. The resources reserved also include the buffers to be used in the transition from the primary to the alternate channel. These transition buffers are reserved at the destination node and are used to ensure that all packets at the destination are passed to the client in the correct sequence. This is needed especially when the delay along the alternate route is less than that along the primary route. The correct handling of resources on all the links that are in common between the primary and alternate routes is another function of the algorithm.

Along a common link there are two methods that can be used to establish an alternate channel. The first method is to reserve sufficient resources for both the primary and alternate channels, and to recover all of the primary channel resources after the transition to the alternate channel. It should be noted that the first method assumes that there are sufficient resources available on this common link to accommodate the primary and alternate channels simultaneously. The second method, applied when there are insufficient resources to accommodate both the primary and alternate channels simultaneously, is to ensure that the more resource intensive channel can be accommodated and then to sequentialize access to this common link, thereby ensuring that all of the primary channel packets traverse the common link before the alternate channel packets. As resource reservation is usually along multiple dimensions, it is probable that no channel is uniformly larger than the other and so resources will be reserved to accommodate the larger resource demand along all dimensions.

The *Transition* algorithm ensures that the transition from the primary to the alternate channel does not violate the client's modification contract. The modification contract guarantees that either there will be no performance violations or that the number of performance violations will be bounded. The "no violation" DCM modification contract assures that there will no performance violation by restricting the value of the alternate route parameters. These restrictions placed on the parameters of the alternate channel ensure that three of the four possible performance violations (i.e. throughput, delay, delay jitter, and packet ordering) will not occur. Packet ordering violations may occur.

[3] In this case jitter is defined as the difference between the delays experienced by any two packets on the same connection.

To ensure that no packet ordering violations occur, a packet ordering mechanism is needed to reorder packets at the destination node. This mechanism uses the transition algorithm. This algorithm determines the time interval that the destination node needs to hold packets on the alternate channel so as to ensure that the correct ordering sequence is maintained without violation any of the delay and delay jitter bounds on the packets of both the primary and alternate channel. With the "bounded violation" contract there are no parameter restriction on the alternate channel parameters, however, the client must be able to accommodate the number of violations specified by the bound.

The *Routing* algorithm determines an alternate route which has the highest probability of successful channel establishment. This is accomplished by using the client's traffic and performance requirements and the current real-time network state. The algorithm also takes into consideration the resources that have been currently reserved for the primary channel. Source routing is achieved by using a modified, constrained, version of the Bellman-Ford algorithm. In this algorithm a directed graph is formed in which the nodes represents switches and hosts and the edges represent links. The weight of the link is the sum of the transmission and propagation delay, and the minimum possible queuing delay at the link. This queuing delay is determined by using the client traffic requirements and the current real-time traffic on the link. A path is chosen by searching paths from the source to destination node, in order of increasing hop count, until a path is found that satisfies the delay and delay jitter bound requirements of the client. At this hop count all paths are examined and the path that has the minimum delay, yet satisfying the delay and delay jitter conditions, is selected. Complete details of the algorithm can be found in the reference [9]. This algorithm seeks to maximize network utilization (by minimizing hop count), to load balance (by choosing among the minimum delay path at that hop count), to minimize the channel establishment time (by limiting the algorithms search space), and to minimize the establishment message rejection probability (by using the traffic and performance requirements of the client and the current network state).

3.3 Network Initiated Adaptation

In network initiated adaptation, the client initially agrees to a range of traffic and performance requirements that are acceptable and a maximum decrease, the step size, that would be permissible upon consent. Based on the network state and other client demands, the network may reduce the level of service provided to some of these clients (with the consent of those clients) so as to achieve a network state with more optimal performance.

An example of network initiated adaptation is the redistribution of fixed resources among channels in a multicast channel session. In this example a multicast channel is currently in session and a new user wishes to join the session. In the case where there are no additional resources available to the multicast channel, possibly due to saturation of the network, the resources needed to admit this new user can be collected from the other multicast channel participants with their consent. The redistribution of these resources among a larger number of participants may result in a reduction of the quality of the service to some participants.

This situation can also occur in the case of errors or failures on links whereby the resources collected, by reducing the resources of certain clients, enable the rerouting of affected clients on these failed links, thus increasing the robustness of the network.

Another scenario of interest is that of resources on demand whereby a client, willing to pay for immediate access, can be granted access to the network by collecting resources from the other willing clients. The consenting client's service can only be reduced to the specified lower threshold value and only in the specified steps. Credits or incentives can be provided to these consenting clients to encourage them to reduce their level of service.

The request/response is also asynchronous here, since the network has to find and collect the resources as soon a client needs them. There will be no performance violations in this type of adaptation.

3.4 Client Initiated Adaptation

In client initiated adaptation, the client explicitly request a change in its channel's performance requirements, and the network honors the request based on the current availability of its resources. This adaptation occurs due to an explicit client request which can be an increase or decrease of the QoS currently provided and is synchronous with the request in that the response takes place within a limited interval after the request. A good example of this is the browsing of images stored in a remote data-bases where the client wishes to double its bandwidth so that it can fast forward or fast rewind its frames. The client requests the additional bandwidth and the network determines if the bandwidth is available. If the bandwidth is available, the network honors the request within the specified time interval and informs the client when the resources have been made available. If the resources are not available, the network sends a denial response to the client.

3.5 A Simple Adaptation Policy

A simple adaptation policy will be discussed in this section and a simulation experiment and its analysis will be provided, in the next section, to illustrate the effectiveness of our proposal. This simple policy, that we called Consenting Equal Division (CED), was devised to illustrate the usefulness of the adaptation services. It divides the resources needed by a new client's request equally (in terms of percentage) among all of the participating clients who consent to adaptation. A participating client is one who has indicated a willingness to participate in network initiated adaptation and has therefore specified its range of traffic and performance requirements and its step size. Only participating clients are asked to modify their resource reservations.

In this policy both network initiated and client initiated services are available. When the policy manager receives a request for the establishment of a new channel or a request for the enhancement of an existing channel, it first determines if it is possible to accommodate the request. The manager decides if the request is to be considered at this time by using a heuristic based on the adaptation rate and the current load in the network. If adaptation can be considered, it queries all of the participating clients to determine which, if any, will consent to adaptation. This query is accomplished by sending a "consent" packet along a multicast tree constructed on the participating clients list. The clients append their responses to the packet and send the packet back to the manager. The clients service cannot be reduced below the threshold specified at the establishment of the service, and will not be reduced more than the step size initially indicated. After the list of consenting clients has been compiled, the manager equally divides the needed resources among the clients, taking into consideration the needed resources, the threshold and the step size of each client, and informs each of the consenting clients of the reduction in service. The selected clients are also informed of reductions to their service using the multicast tree mechanism mentioned above. The actual resource reservation reduction only occurs when the clients return an acknowledgement of the reduction message to the manager. This enables the client to reduce its resource consumption before the reserved resources are reduced. The manager then establishes the new channel or enhances the performance of the existing channel using the newly acquired resources. It should be noted that this equal (percentage) division may not result in equal resources recovered from the client as the step size and the discrete nature of some of the resources will not permit it. However, these recovered resources are divided as evenly as possible across all consenting clients.

As clients terminate their sessions, the resources recovered by the network are not restored to the original clients unless explicitly requested. Current clients must use the client initiated service to recover their previous quality of service levels. A client may ask for more resources than was previously released by that client. The manager looks at these enhancement requests and the new client requests and determines which request, if any, is accepted. In this policy, priority is given to current clients over new clients, and the longer the duration of the client's channel the higher its priority.

This simple adaptation policy was designed as a prototype to be implemented on a local area FDDI real-time network, at Berkeley, and it is intended as an investigative tool rather than a definitive solution to the problem of dynamic channel management. The prototype seeks to exercise the DCM algorithms by using these adaptation mechanisms under an useful management policy on a production network. It should be noted that for any dynamic adaptation scheme there is the problem of stability caused by the oscillations generated by the changing network state and the client demands. This oscillation problem is to be addressed in future work.

4 Simulations and Results

In this section, we describe the simulation experiment that was conducted on the policy presented in the previous section and provide an analysis of the results. The simple experimental network that was used is shown in Figure 1. In this network all of the links and nodes were homogeneous with the links having a bandwidth of 1 MBps and propagation delay of 10 ms. Links are also bidirectional. Fifteen real-time channels were present during the simulation. There were two types of adaptive services, A and B, where their services, for the purpose of this experiment, are differentiated only by throughput. The range for this parameter was 100-50 KBps for the channels of type A and that of 200-100 KBps for channels of type B. The step size for channel types A and B were 20 KBps and 40 KBps, respectively, the delay bound was 80 ms and a packet length of 1 Kbyte was assumed. For the purpose of the simulation experiment, all channels transmit at a constant rate that is the maximum permitted by their reserved resources.

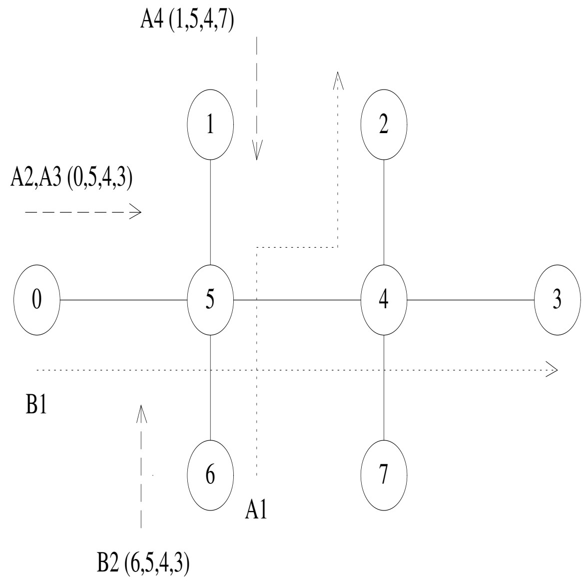

Figure 1: Experimental Network

In this simulation experiment there were 6 adaptive channels present in the network, 4 with type A service, and 2 with type B service. There were 9 other real-time channels present in the network that were not adaptive but routed such that the link between nodes 5 and 4 was saturated throughout the duration of the experiment. Initially all adaptive channels were given the maximum throughput in their range. The six adaptive channels are shown in Figure 1. The channels with type A service are numbered A1, A2, A3, and A4 and the channels of type B service are numbered B1, and B2. In this experiment, all adaptive channels with the same type of service act as a group whereby they all give the same answer (i.e. *consent* or *rejection*) to a request for service adaptation. As all channels in the group show the same throughput performance only one channel from each group, A1 and B1, will be graphically depicted. The routes taken by A1 and B1 are shown by dotted lines in Figure 1, while the routes taken by the other channels are presented in parenthesis where the values within the parenthesis, going from left to right, are the nodes encountered going from the source to the destination.

The simulation can be divided into four time intervals. The four intervals correspond to the intervals from 0 to 2500 secs, 2501 secs to 5500 secs , 5501 secs to 8000 secs, and 8000 secs to

10000 secs. Within each period there is a *transition sub-period*, where there is a transition from the old bandwidth reservation to the new bandwidth reservation, followed by a *stable sub-period* corresponding to transmitting at the new reserved rate. During the first period there were no new client request or enhancements by existing clients. Figure 2 and 3 illustrate this situation. Figure 2 is a graph of *Throughput vs simulated time* for client A1, while Figure 3 is the same graph for the client B1. The throughput is a rate that is measured in packets per interval where the interval is 1 second. During the first period the graphs indicate that the maximum possible throughput for each channel was actually provided by the network, according to their initial specifications.

At the start of the second period (i.e from 2501 secs to 5500 secs) a new client requested a channel with type B service along the route 6, 5, 4, 2, and the policy manager attempted to establish this new connection. A *network initiated* adaptation was attempted as the manager queried all of the participants and all 6 adaptive channels consented to adaptation. The throughput resource was divided among all of the clients, (a reduction of 20% for all current clients) both the current clients and the new client, and the new client was given the same service as that of other clients within its service type group. This reduction is shown in Figure 2 with the decrease from a reservation of 100 packets per interval to a reservation of 80 packets per interval. In Figure 3, the decrease was from 200 packets per interval to 160 packets per interval. The time taken by the manager to achieve adaptation is the sum of the consent message round trip time, the reduction message round trip time, the maximum reduction message acknowledgement time among the consenting clients, and the maximum time taken to modify the resources of the consenting channels. In this period the adaptation time was 306 ms and the establishment time of the new channel was 66 ms. It should be mentioned that both the consent and reduction message follow a minimum spanning tree route that encompasses the participating and consenting clients, respectively.

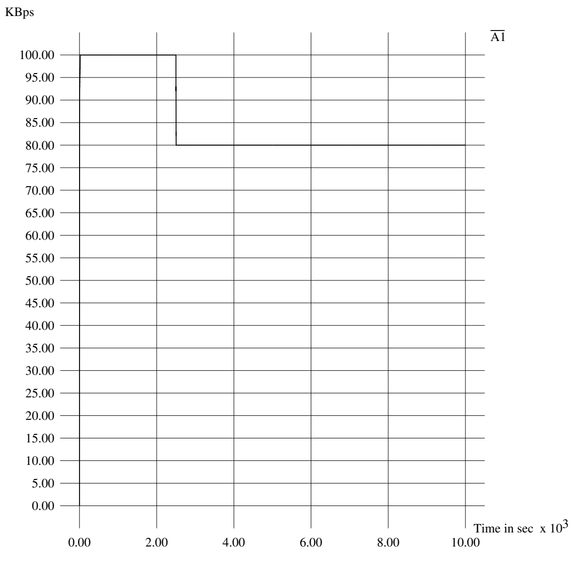

Figure 2: Throughput vs time - Client A1

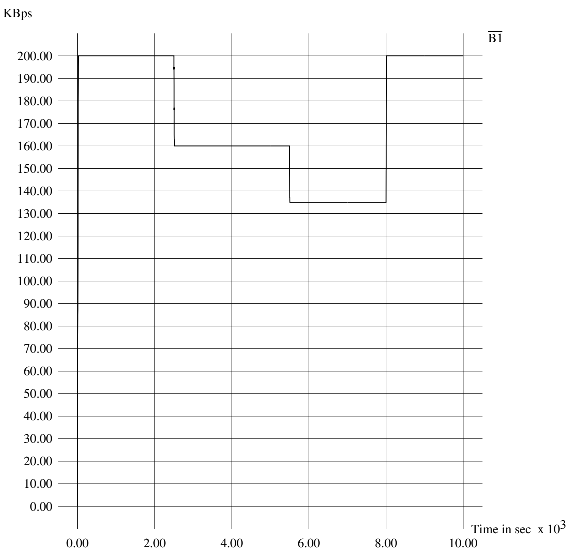

Figure 3: Throughput vs time - Client B1

In the third period (i.e. from 5501 sec to 8000 sec) a new request was made for a channel with service type A along the route 0, 5, 4, 3 , and the manager queried the participating clients for consent. All clients with channels of service type A refused to adapt while all clients with channels of service type B gave their consent. These 3 service type B clients had their service reduced by 17 % and the new channel was admitted with its throughput equal to that of the other channels with service type A. As the clients with service type A channels did not consent to adaptation, Figure 2 did not show any change in throughput. The clients with channels of service type B, which did consent, show a decrease in 17% to 133 packets per interval as shown in Figure 3. The adaptation time during this period was 275 ms and the establishment time of the new type A channel was 64 ms.

In the fourth period (i.e. from 8001 sec to 10000 sec) a channel with service type B (with the current throughput reservation of 133 KBps) along route 6, 5, 4, 3, and a client with a service type A channel along route 0, 5, 4, 3 (with throughput 80 KBps) terminated their sessions. Also at the start of this period, clients B1 and B2 attempted a *client initiated* adaptation by requesting an enhancement in their throughput to 200 KBps. This was accommodated by the policy manager. Figure 2 shows no change as the clients with service type A did not choose to enhance their service. Clients B1 and B2 wished to recover their original throughput and this is shown in Figure 3. As there was no need to reduce the resources of current clients the adaptation time is 0 ms while the modification times of the 2 type B channels were 67 and 65 ms, respectively. Table 1 shows the maximum delay experienced by packets along the channel of each of the six participating clients. The table shows that there were no delay performance violations as the maximally delayed packets traversing any of the channels were within their delay bounds. There were also no packet ordering violations recorded for any channel. This is important as it verifies that adaptation in no way violated the delay or packet ordering performance guarantees of existing channels, thus there was no disruption of service to any of the real-time channels.

Client	*Maximum Delay*	*Delay Requested*
A1	72	80
A2	73	80
A3	72	80
A4	73	80
B1	74	80
B2	73	80

Table 1: Maximum Delay Experienced vs. Delay Requested - All Clients

5 Conclusion

To improve the efficiency and flexibility of guaranteed performance services in integrated services networks, we presented an approach that allows the graceful adaptation of guaranteed performance services needed to support dynamicity of the network load and client requirements.

This graceful adaptation can be initiated either by the client or the network. In both cases, the network modifies the connections with the client's consent and guarantees a bounded or no performance requirement violations subject to a pre-negotiated modification contract with the client.

An adaptation policy is used to provide the control for client and network initiated adaptation which are both based on the Dynamic Connection Management (DCM) scheme. This policy should allow a more efficient use of the network's resources by performing cooperative, consenting, high-level multiplexing of real-time connections. A simple policy has been devised and a simulation experiment was performed to show the positive effects of graceful adaptation.

The results presented are encouraging and illustrate the feasibility of our approach. This initial investigation has shown that the main overhead due to the policy, the adaptation delay, is within acceptable bounds (i.e. 0.25 to 0.31 sec) when balanced against the increased access and network utilization. In all cases, there was no disruptions of service (in the form of performance violations) to any of the real-time connections on the network.

In future, we plan to investigate a number of more complex adaptation policies. There are many issues that we are currently addressing such as providing distributed control while preventing oscillations in the network, determining the effect of graceful adaptation over wide area networks, and reducing the adaptation time. We are also currently implementing the DCM schemes and the adaptation policy on a local area testbed for more intensive investigation.

References

[1] David P. Anderson, Ralf Guido Herrtwich, and Carl Schaefer. SRP: A resource reservation protocol for guaranteed performance communication in internet. Technical Report TR-90-006, International Computer Science Institute, Berkeley, California, February 1990.

[2] David Clark, Scott Shenker, and Lixia Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proceedings of ACM SIGCOMM'92*, pages 14–26, Baltimore, Maryland, August 1992.

[3] D. Ferrari, J. Ramaekers, and G. Ventre. Client-Network Interactions in Real-Time Communication Environments. In *Proceedings of HPN'92, International Conference on High Performance Networking*, Liege, December 1992.

[4] Domenico Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, November 1990.

[5] Domenico Ferrari, Anindo Banerjea, and Hui Zhang. Network support for multimedia: a discussion of the Tenet approach. Technical Report TR-92-072, International Computer Science Institute, Berkeley, California, October 1992. Also to appear in *Computer Networks and ISDN Systems*.

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

[7] Michael Gilge and Riccardo Gusella. Motion video coding for packet switching networks – an integrated approach. In *SPIE Visual Communications and Image Processing '91*, November 1991.

[8] C. Parris, S. Keshav, and D. Ferrari. A framework for the study of pricing in integrated networks. Technical Report TR-92-016, International Computer Science Institute, Berkeley, California, March 1992.

[9] Colin Parris and Domenico Ferrari. A dynamic connection management scheme for guaranteed performance services in packet-switching integrated services networks. Technical Report TR-93-005, International Computer Science Institute, Berkeley, California, January 1993.

[10] Y. Tobe, H. Tokuda, S. T.-C. Chou, and J. M. F. Moura. QOS Control in ARTS/FDDI Continuos Media Communications. In *Proceedings of ACM SIGCOMM 92*, pages 88–98, 1992.

[11] Claudio Topolcic. Experimental internet stream protocol, version 2 (ST-II), October 1990. RFC 1190.

[12] Nanying Yin and Michael G. Hluchyi. A dynamic rate control mechanism for integrated networks. In *Proceedings of INFOCOM'91*, 1991.

[13] Lixia Zhang. *A New Architecture for Packet Switched Network Protocols*. PhD dissertation, Massachusetts Institute of Technology, July 1989.