INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

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 94704-1105, USA.

TR-93-011
March 1993

Abstract

Most of the solutions proposed to support real-time 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 before the start of the application on the basis of resource availability and real-time network load at that time, and are usually kept for the duration of the application. However, such an approach shows two major limitations: first, the communication service provided is usually fixed, with no or limited capability of adaptation 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 adaption of guaranteed performance service connections. Mechanisms have been devised to allow changing of the traffic and performance parameters of a real-time communication during its lifetime. These mechanisms, together with an adaption 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

*This research is 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, Digital Equipment Corporation, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute.

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

examples for both types and we also present results from simulation experiments to verify the correctness our proposal.

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 [5].

Several solutions have been proposed to support guaranteed performance service communication in packet-switching networks [1, 7, 12, 14]. They all adopt a connection-oriented and reservation-oriented 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. 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 people join the conference it may not be feasible to support all the people with the same good quality, due to resource constraints, as when there are only two participants. It may be more desirable to degrade the quality of the established connections rather than to deny newcomers to participate to the conference.

In this paper, we present a scheme to dynamically manage the quality of service (QoS) of real-time communications by changing the parameters that specify it during the lifetime of the connection. The parameter change can be initiated either by the network, or by the client. In the first case, a client should specify, at the time of the request of a communication, 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), with respect to the current load in the network.

Parameter changes can also be initiated by clients to reflect the dynamicity of the client 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.

Our scheme of dynamically managing communication parameters is within the framework of 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 guarantee 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 change from the old quality of service to the new with no or limited disruption. This has been accomplished by introducing a mechanism we called *graceful adaptation of service*.

The paper is organized as follows: in Section 2 , the motivation for this work is discussed. 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 activated by the clients. These mechanisms are all based on the Dynamic Connection Management (DCM) [9] algorithms, which are extensions of the Tenet [7] 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 Motivation

The problem of providing a flexible, real-time communication service, suitable for a wide range of multimedia applications, is still to be completely solved in the framework of guaranteed quality of service.

In fact, we believe that this problem is characterized by two different aspects. The first one relates to how the client can interact with the network in order to get a communication service that matches its needs. In the real-time network prototypes proposed in the literature, clients have very limited or no capabilities of negotiating the traffic and performance characteristics of a connection with guaranteed quality of service. In addition, once a connection has been established, the QoS provided to the client usually cannot be changed.

The second aspect relates to how the network can react to the appearance of particular traffic situations. For example, depending on the evolution of the clients' demand for real-time communications, saturation of the resources for real-time traffic might occur in some areas of an internetwork, such that the establishment of new real-time connections passing through that area is dramatically limited. In other cases, the occurrence of failures might

require proper adjustment in the traffic distribution in order to properly solve the problem, for example by rerouting the traffic affected by it.

The issues of adaptation of a communication service and, more generally, of dynamic management of real-time connections, have only been partially addressed in the literature.

The proposals presented in [10] and [1] are characterized by a model where the interaction between a real-time network and its clients is minimal, and is limited to the establishment phase of a connection.

In the ST-2 protocol [12], 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.

A more general approach to assured quality of service is proposed by the Tenet scheme for real-time communication [7]. The client specifies its throughput requirements in terms of a minimum and an average inter-packet arrival time, the latter averaged over a client-specified interval. The delay requirements can be specified in terms of an absolute upper bound on the transmission delay (*deterministic service*), or of a probabilistic bound (*statistical service*), or as an upper bound for the delay jitter (*bounded-jitter service*) [5]. In terms of the client-network interaction, the current implementation of the scheme supports only partially flexibility and adaptivity of the network. Even though a connection administration protocol is expressly included in the protocol stack [2], the capability of a client to dynamically interact with the network is still limited.

Recently, two different solutions have been proposed to increase the adaptivity of real-time networks. In [3], 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 measurements. Since in the 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 [11], the user can specify the minimum and maximum quality 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. Once 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. Another limitation is that it is only discusses in a local area network environment, issues associated with a more general internetworking environment are not addressed.

In the context of support 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 [8, 13]. Their proposals differ from ours in that they assume a connectionless network and thus no performance guarantees are provided.

The approach we propose in this paper is to improve the network flexibility by taking advantage of the adaptive nature of some applications that still require guaranteed performance services, so that we increase the benefit to both the network and the client. One problem associated with guaranteed performance service is that it requires very strict resource management and may result in lower utilization of the network of real-time traffic. To alleviate the problem, statistical guarantee service, in addition to the deterministic guarantee service, can be offered, to exploit the traffic burstiness and obtain statistical multiplexing gain [7]. Another way of addressing the problem is what we call *cooperative, consenting, high-level multiplexing*. This is achieved by allowing connections to adapt or modify their performance parameters in light of changes in clients' requirements and network state.

It is our opinion that two types of adaptations are needed: a *network initiated* adaptation and a *client initiated* adaptation. In network initiated adaptation, the network uses the client and the network state to determine if availability can be enhanced by a reduction in the resources of a subset of the clients, who have specified their willingness to this adaptation at the moment of the request of the service. This adaptation can be exploited either at the establishment of the connection, if the network is not able to accommodate the new request according to the optimal values expressed by the client, or during the connection lifetime. In the case new resources will become available, they will be granted to the connection in order to provide him the quality of service originally requested.

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 QoS is under the client's control and is within well specified ranges. The second is that those applications will appreciate this flexibility as it provides a higher probability of having the communication service established by the network [4], and possibly also a pricing scheme advantageous to the client.

The other approach, the client initiated adaptation, allows a client to request a better quality of service, in terms of improvement of some or all of the performance parameters. This is the case, for example, when a client may need more resources as its adaptive application may encounter situations where compression is reducing the quality of service below tolerable limits. The same mechanisms might be feasible also in the other sense, to allow the reduction of the communication performance due, for example, to a reduction in the QoS requirements of an application. The solution we propose is to extend the client/network interaction from the establishment time to the whole lifetime of the connection.

We believe that the introduction of these two different adaptation mechanisms are beneficial for both the client of real-time communication services, and the network. For the former, these mechanisms should guarantee an improvement in the establishment of a communication and an increased control over the allocation of the resources to its needs, for example to have control over the costs. For the latter, the high-level multiplexing introduced by these mechanisms should guarantee an improvement in the overall network utilization of the resources for real-time traffic and, possibly, an increase in the revenue.

3 Graceful Adaptation Schemes

In this section, we present the mechanisms that provide the channels[1] 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 schemes, network-initiated and client-initiated, will then be discussed. A simple adaptation policy will also be presented in the final subsection.

[1] There are different terms in the literature for the same or similar objects. A channel is also called a connection, a circuit, a conversation, a flow, etc..

3.1 The Tenet Model

In the Tenet Scheme, a guaranteed performance service connection (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 [6].

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 - the averaging interval.
- 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.[2]
- Z - the probability that the delay of the packet is smaller than the delay bound, D.
- 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 of each packet are met at that node, and the client specified end-to-end performance guarantees are thus satisfied.

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

3.2 DCM Algorithms

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 limited amount of time called *transition interval*.

The DCM scheme consists of three algorithms:

- The *Channel Administration* algorithm.
- The *Transition* algorithm.
- 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 particularly needed 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 from the primary 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 both the primary and alternate routes. The second method is applied when there are insufficient resources to accommodate both primary and alternate channel along this common link. In this case, the channel administrative algorithm reserves enough resources to ensure that the more resource intensive channel can be accommodated and then it 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 many 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. Of the four possible performance violations, (throughput, delay, delay jitter, and packet ordering) alternate channel parameter restrictions are sufficient to ensure no violations in three of them; however, to ensure no packet ordering violations a packet ordering mechanism is needed to reorder packets at the destination node. This mechanism is the transition algorithm. The algorithm holds 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.

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 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 call blocking), load balance (by choosing among the minimum delay path at that hop count), while minimizing the channel establishment time (by limiting its search space for a proper routing), and the establishment 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 is acceptable and the step size increase or decrease 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. As the resources are fixed, possibly due to saturation of the network, they need to be collected from the other participants, with their consent, in order to admit this new client. Obviously the recovery of these resources would reduce the quality of the service of each client accordingly.

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 clients 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 clients to accept 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 scheme.

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 video frames 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 request to the client.

3.5 Adaptation Policy

A simple policy will be discussed in this section while simulation experiments and their analysis are provided in the next section to validate our proposal. This simple policy, that we called Consenting Equal Division (CED), was devised to illustrate the usefulness of the adaptation mechanisms. It divides the resources needed by a new request equally (in terms of percentage) among all of the clients participating in adaptation who consent to it. Hence, in order to allow the network to include its name in the list of the clients available for the adaptation, a client needs to be participating in the adaptation and to consent to this reduction in service.

In this policy both network-initiated and client-initiated adaptation is possible. When it is received the request of a new client for the establishment of a service or of an existing one for the enhancement of its service, the manager (who implements the policy) determines in two passes if it is possible to accommodate the request. In the first pass, the manager queries all of the clients to determine which of the participating clients consent to adaptation. This first pass is accomplished by sending a "consent" packet along a multicast tree constructed on the participating client list. The clients append their responses to the packet and send the packets back to the manager. The clients are also informed of reduction using a similar

mechanism. The clients service cannot be reduced below the threshold specified at the establishment of the service, and will only be reduced by 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. 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. Finally, the last pass consist in the modification of the selected channels according to the new resource reservations.

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

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. As it was the intention of this simple policy only to illustrate the feasibility of improving the flexibility of guaranteed performance communication services, the stability problem has not been addressed.

4 Simulations and Results

In this section, we describe the simulation that was done on the policy presented in the previous section and provide an analysis of the results. The simple experimental network we used is shown in Figure 1. In this network all of the links and nodes were homogeneous with a bandwidth of 1 Mbps and propagation delay of 10 ms. Fifteen real-time channels were present during the simulation. There were two types of adaptive services, A and B, where the services are differentiated only by throughput. The range for such parameter was 100-50 Kbps for the channels of type A and that of the channels of type B was 200-100 Kbps. For both the types the step size was 10 Kbps, the delay bound was 60 ms and a packet length of 1 kbyte was assumed. All clients with type-of-service A and B agreed to participate in channel adaptation.

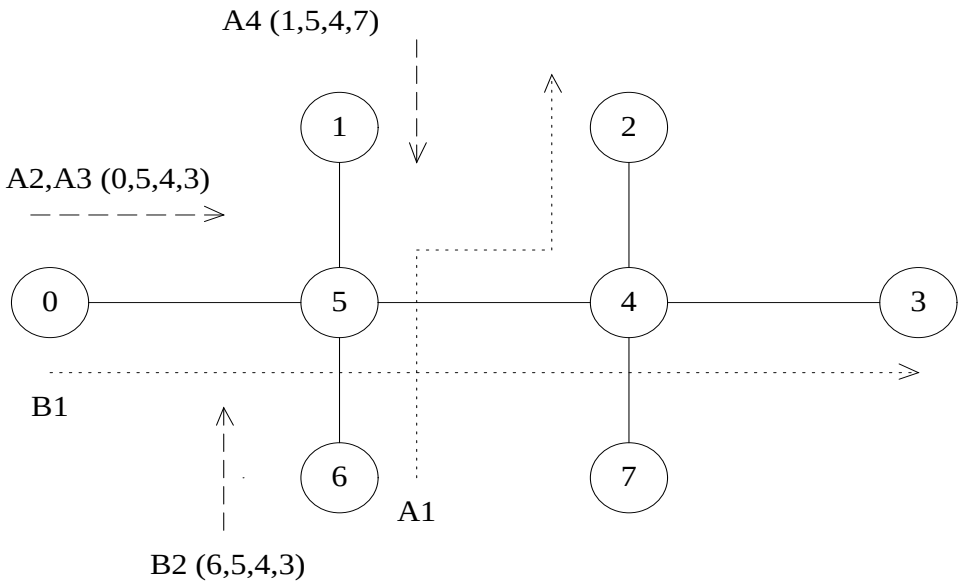

Figure 1: Experimental Network

In this simulation there were 6 adaptive channels present in the network, 4 with type-of-service A, and 2 with type-of-service type B. The other real-time channels present in the network were not adaptive and the link between nodes 5 and 4 was saturated throughout the duration of the simulation. Initially all channels were given the maximum throughput in their range. The six channels are shown in Figure 1. The channels with type of service A are numbered A1, A2, A3, and A4 and the channels with type of service B are numbered B1, and B2. In this simulation, all channels with the same type-of-service act as a group whereby they all give the same answer, consent or rejection, to a request for service adaptation. As all channels in the group show the same throughput performance and have very similar delay histograms, 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. It should be noted that this network was chosen to isolated the effects of the adaptation schemes.

The simulation can be divided into four time intervals. The four intervals correspond to the intervals from 0 - 2500 , 2501 - 5500 s , 5501 s - 8000 s, and 8000 s - 1000 s. In the first period there were no new client request or enhancements by existing clients. Figure 3 and 4 illustrate this situation. Figure 3 is a graph of *Throughput vs simulated time* for

client A1, while Figure 4 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 the initial specifications.

At the start of the second period a new client requested a channel with type-of-service B 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 determined that all participants consented to adaptation. The throughput resource was divided among all of the clients, (a reduction of 20% for all current clients) both current and new, and the new client was given the same service as that of other clients within its type-of-service group. This reduction is shown in Figure 3 with the decrease from a reservation of 100 packets per interval to a reservation of 80 packets per interval. In Figure 4, the decrease was from 200 packets per interval to 160 packets per interval.

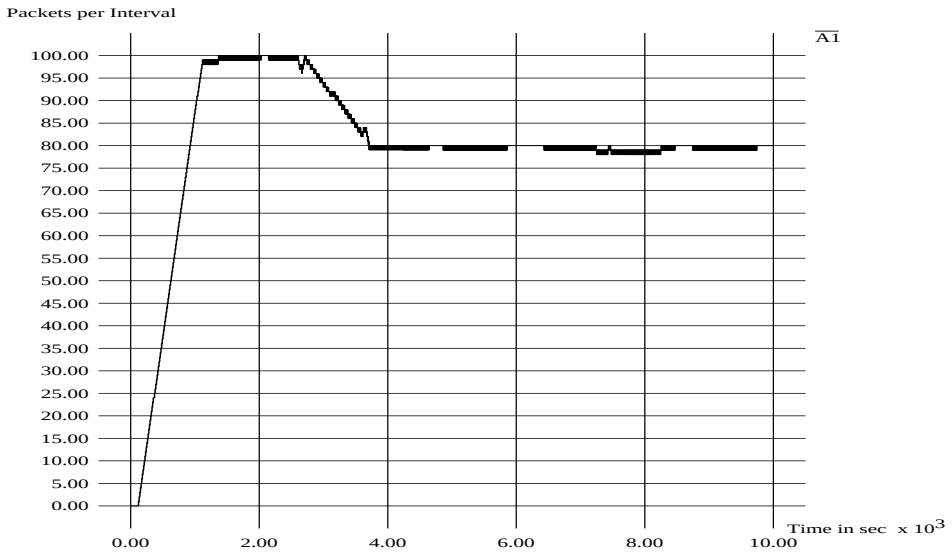

Figure 2: Throughput vs time - Client A1

In the third period a new request was made for a channel with type-of-service A, and the manager queried the participating clients for consent. All clients with channel of type-of-service A refused to adapt while all client with channels of type-of-service B gave their consent. These 3 clients had their service reduced by 18 % and the new channel was admitted with its throughput equal to that of the other channels with type-of-service A. As the clients with type-of-service A channels did not consent to adaptation, Figure 3 did not

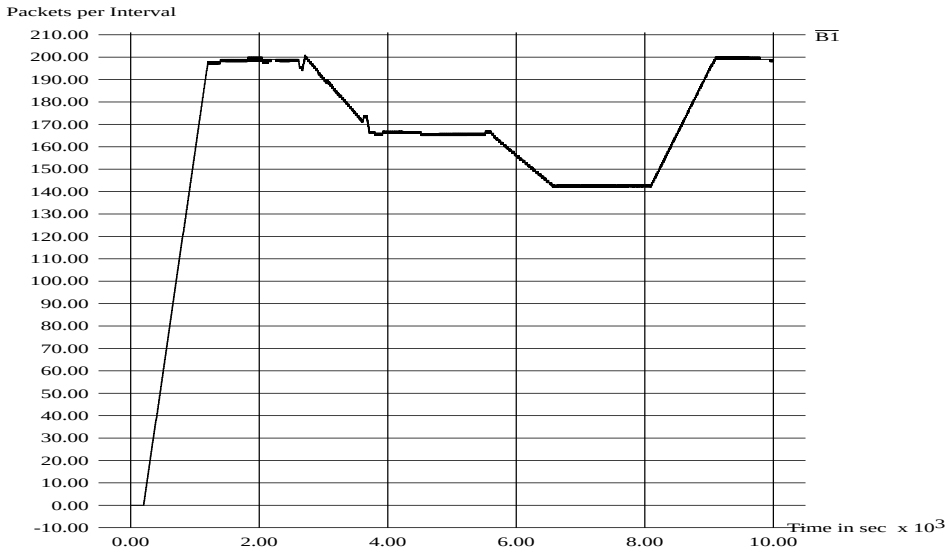

Figure 3: Throughput vs time - Client B1

show any change in throughput. The clients with channels of type-of-service B, which did consent, show a decrease in 17% to 130 packets per interval as shown in Figure 4.

In the fourth period a channel with type-of-service B (with the current throughput of 130 Kbps), and a client with a type-of-service A channel (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 3 shows no change as the client with type-of-service A did not choose to enhance their service. Clients B1 and B2 wished to recover their original throughput and this is shown in Figure 4. The delay histograms in Figure 5 and 6 illustrate that there were no performance violations in channels A1 and B1 for the whole duration of the experiment.

Table I shows the maximum delay experienced by packets along the channel of each of the six participating client. The Table shows again that there were no delay performance violations as the maximally delayed packets traversing any of the channels were within their delay bounds. All packets were also checked to determine if the guarantee over packet-ordering had been satisfied by the network and no out of sequence packets were found.

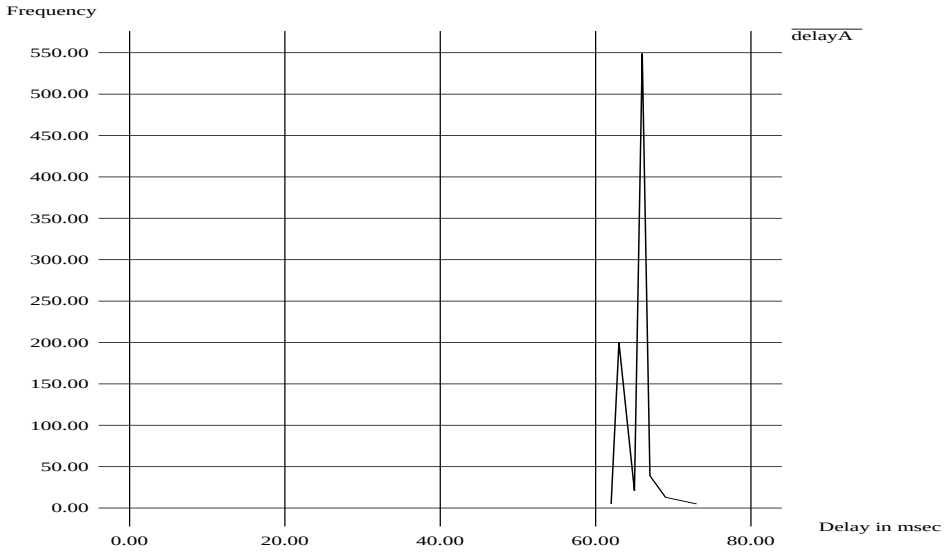

Figure 4: Delay Histogram - Client A1

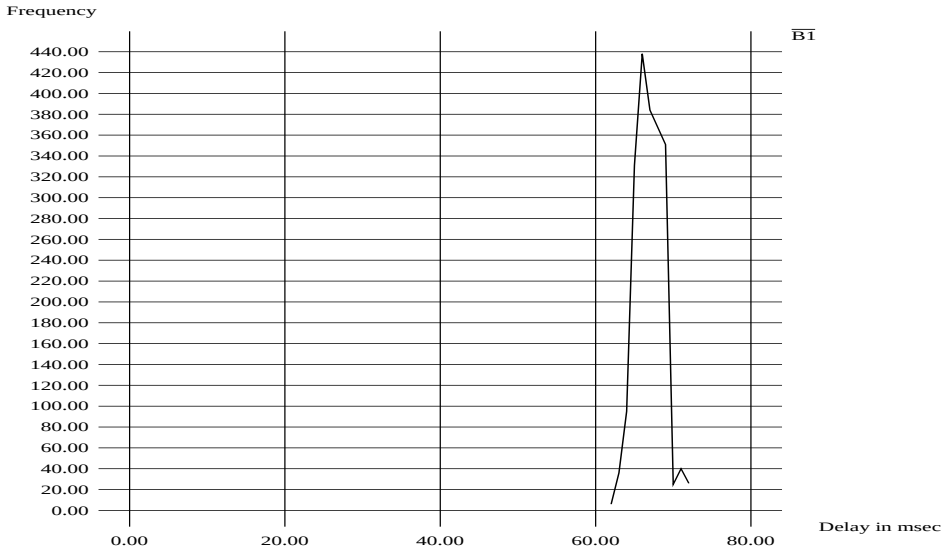

Figure 5: Delay Histogram - Client B1

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 of real-time protocols providing quality of service guarantees in communication, we proposed a mechanism called graceful adaptation of service. In this mechanism, a client and the network are capable of adapting the characteristics of a real-time connection to the changes in their requirements. In both cases, however, the network still commits to provide a guaranteed performance with no or limited disruption in the communication.

The mechanism proposed, if integrated with a proper adaptation policy, can allow a more efficient use of the network resources, by performing a cooperative, consenting. high-level multiplexing of the real-time connections. A simple policy has been devised to show the positive effects of our proposal, and with this policy simulations of dynamic, graceful adaptation of real-time connections have been executed.

The results presented are very encouraging and show the feasibility of the scheme. In future, we plan to evaluate a number of adaptation policies, in order to analyze their fitness for both the clients and the networks. In addition to that, we are interested in evaluating the impact over the stability of a network of the introduction of adaptation mechanisms like the one we presented.

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] Anindo Banerjea and Bruce Mah. The real-time channel administration protocol. In *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 160–170, Heidelberg, Germany, November 1991. Springer-Verlag.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[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] G. M. Parulkar and J. S. Turner. Toward a Framework for High-Speed Communication in a Heterogeneous Networking Environment. *IEEE Network Magazine*, March 1990.

[11] 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.

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

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

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