Client-Network Interactions in Quality of Service Communication Environments[1]

Domenico Ferrari, Jean Ramaekers[2] and Giorgio Ventre

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.

Abstract

Multimedia communication, with its strong requirements for high speed, assured quality, and reliable networking, is stimulating a great research effort towards the development of real-time protocols. Some protocols of this type have been proposed, which offer communication services with different levels of commitment in providing quality of service guarantees. In this paper we study the feasibility of an extended client interface that allows more flexibility in the client-network interactions. The proposed model improves the utilization of network resources, and increases the network's capability to support multimedia traffic, while continuing to offer a guaranteed quality of service.

Keyword Codes: C.2.0; C.2.1; C.2.2
Keywords: Computer-Communication Networks, General; Network Architecture and Design; Network Protocols

1. Introduction

Multimedia applications, with their strong requirements for high-speed and high-quality communication services, have stimulated the development of new network architectures and protocols. One of the most important problems to be solved is how to provide network clients with an improved quality of service (QoS); this does not only requires higher bandwidths, low delays and low loss rates, but also guaranteed upper or lower bounds for these performance indexes.

Any solution to this problem will necessarily imply a move from the traditional, best-effort kind of service, currently provided, for example, by the Internet architecture, to a new kind of communication infrastructure. New models for internetworks are required,

[1]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, Digital Equipment Corporation, 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.

[2]On sabbatical leave from University of Namur, Belgium

where traffic with different communication requirements will be differently managed, so that applications with stronger needs will be privileged over and protected from the usual data traffic that floods today's computer networks [2].

Several protocols have been presented in literature that allow a client to obtain communication services with guaranteed performance. Most of the proposed solutions are based on a communication abstraction consisting of a connection with specified performance characteristics. We shall call this abstraction a *channel*. The establishment of a channel and the resources to be associated to it are controlled by the network through the us of an admission policy. In some schemes, the admission of a new channel depends on the availability of a sufficient amount of resources in the network nodes, to prevent the new channel from jeopardizing the performance of the already established connections.

But what kinds of communication requirements do multimedia applications have? A precise answer to this question is very difficult, due to the high number of these applications that have been and are being proposed in the literature [7]. They range from home video distribution and distributed education, to electronic collaboration (CSCW) and remote system control. While the first two applications, albeit very demanding in terms of bandwidth, can admit some degradation in the reliability of delivery and in communication delays, the latter two impose stronger requirements on the overall quality of the communication service.

Project Sequoia 2000 and the Bay Area Gigabit Network can be considered as two interesting experiments involving multimedia communications with very demanding but different requirements. Project Sequoia 2000 is a project jointly sponsored by the Digital Equipment Corporation and the University of California and aimed at proposing new techniques for solving research problems related to a new scientific discipline, earth system science. The major goal of this discipline is to develop an integrated approach to observing the Earth, and to devise remedies to the negative impact of human activities on the global environment. This requires the availability of a large amount of data produced by researchers in a number of disciplines, such as meteorology, bioclimatology, oceanography, hydrology and geochemistry [10].

From the networking point of view, this project poses a number of challenging problems, like determining effective ways of visualizing satellite data. An example of these problems is to allow the fast-forwarding in either the temporal or spatial dimension of a composite sequence of frames collected from a weather satellite and stored in a remote file server. The goal is to have an effect similar to that achieved by TV weather forecasters, who show movement of a storm by composing a sequence of images collected in a 24-hour period. A single frame from a satellite instrument contains 4 Mbytes, assuming 2K x 2K pixels and 1 byte/pixel for color; we need a broadband network that can not only deliver the 640 Mbits/sec required by a 20 frames per second rate, but also guarantee that the delivery of these data will not suffer any pause or strong delay that would degrade the real-time viewing.

On the other hand, the Bay Area Gigabit Network will be a testbed for experimenting with applications such as teleseminars, teleconferencing and other forms of electronic interaction among a large number of individuals from research and industrial organizations located in the areas surrounding the San Francisco Bay. For these applications, the bandwidth requirements are expected to be much lower (i.e. 50 Mbits/sec for a single video channel), but with stronger delay and delay variation requirements. In fact, for this kind of application it is very important to achieve not only a low communication delay, but also a good synchronization among the multimedia streams received by all the participants, to allow interactions to be as similar as possible to those achievable, for example, in a traditional seminar [12].

We believe that, regardless of their differences, all these new applications require networks capable of providing a *guaranteed* quality of service. This means that, even if less critical applications, like for example distributed video advertising, may actually ask for a quality of service lower than the one needed by a medical information system, both these applications will require from the network provider a communication service with well defined quality indexes, such as those related to performance and reliability of the communication.

Several proposals for the design of such a service have been recently presented in the literature, and will be briefly described in the next section. However, the design of most of these proposals suffers of a very primitive and inflexible interface between the client and the service.

The problem of providing a flexible real-time communication service, suitable for a wide range of multimedia applications, can be helped by improving the adaptivity of the reservation and control mechanisms of real-time protocols. In this paper, we present a client interface model that can improve the flexibility and the adaptivity of real-time communication networks, and describe how we plan to use it for the next version of the Tenet protocol suite, developed by the Tenet Group at the International Computer Science Institute and the University of California at Berkeley. The proposed model improves the utilization of the network resources, and is expected to increase the network's ability to support real-time channel, while continuing to offer a guaranteed quality of service.

2. Quality of Service Characterization

In spite of the several research efforts made in recent years to provide quality of service in real-time networks, there is still no wide consensus on what should be the features and mechanisms of such networks.

From the client's point of view, the quality of service of a real-time network is determined by the values of some end-to-end performance parameters. Such parameters may be expressed in terms of three quality indexes:

1. *traffic throughput*: this index is related to the amount of data that will be sent through the network per unit time. It usually specifies the traffic communication needs in terms of the bandwidth required. These needs may be specified, for example, as a packet transmission rate or as an interpacket distance. These values may refer to the peak performance or to an average over a specified interval. In addition to representing a requirement of the client, a throughput specification can be used by the network to determine the amount of traffic that will be produced by that particular client. In this sense, this index may be interpreted by the network as a client's pledge to obey certain traffic restrictions.

2. *transmission delay*: this quality index is the delay that the transmitted data will suffer through the network. This parameter may be expressed in terms of an abso-

lute bound or of a probabilistic one. In addition to a delay bound, a bound on the delay variation, or jitter, can also be specified. Depending on the real-time protocols being used, bounds on communication delay and jitter may be either client-specified quality requirements or a measure, evaluated by the network, of the current performance of the real-time service. In the first case these indexes are used by the network to evaluate the feasibility of a connection and the amounts of resources required for its establishment. In the second case they are the result of an evaluation of the capabilities currently available in the network, and are submitted to the client who can accept or refuse the proposed QoS.

3. *transmission reliability*: this quality index is primarily related to the buffering mechanisms involved in data transmission along the network. In packet-switching networks, packets are received by intermediate nodes, and, until they are transmitted to the next node on the communication path, are stored in buffers. Because of the limited size of these buffers, it might happen that, due to traffic congestion, overflows cause some packets to be lost. A probabilistic bound on such kind of losses can be used as a measure of the reliability of a communication service; its value will influence the amount of resources required for the establishment of a connection. In a way similar to what happens for the delay indexes, transmission reliability can be a client-specified parameter, or a parameter measured by the network[1].

The above mentioned indexes can be used alone or in various combinations as quality of service specifiers. As we have seen, in some cases they are specified by the clients, while in other cases they are evaluated by the network and proposed to the client for approval. In all cases they should be considered as representing a commitment by the network to provide a transmission service conforming to given performance parameters. However, a crucial problem is still unresolved in the area of real-time communication networks: how strong this commitment has to be, and what kinds of guarantees a client should receive from the service provider.

The majority of the solutions proposed during the last few years try to solve this problem by using one of two different approaches:

- *Hard Guarantees*: the network commits to offer a service, whose quality is precisely specified through a number of traffic and performance parameters. A *contract* is required between a client and the network, in which the former specifies the characteristics of its traffic and makes a pledge to respect them for the duration of the contract, while the latter promises to provide a service conforming to the client's requirements. To protect the guaranteed level of quality from the effects of misbehaving clients and fluctuations in the network's load, mechanisms for resource reservation, traffic admission control and rate control have to be enforced. Service is not provided if the available resources and the traffic characteristics of the client do not allow the expected service to be achieved.

- *Soft Guarantees*: in this case, when a client requests a real-time service, the network evaluates the current traffic load and the characteristics of the traffic of the

[1] Another aspect of reliability is related to the occurrence of permanent or temporary failures in the network, but it will not be considered in this paper

requesting client. Such characteristics do not have to be related to the worst-case situation as in the hard guarantees approach, but can be descriptive of the average behavior of the client. With these data, the network evaluates an achievable level of quality and submits it to the client for approval. Since this level depends on the assumption that the network's load will conform to the current situation, and that the existing and future clients will not modify it, no strong commitment can be made by the network. In fact, whenever the load conditions of the network change, the quality of service provided changes as well; this calls for adaptive clients, i.e. clients capable to tolerate and compensate fluctuations, and even disruptions, in the network's service.

In the SRP protocol [1], clients can specify their requirements by using three traffic parameters and one performance index. The first three are respectively the maximum message size, the maximum message rate, and the maximum burst size. The performance index is a delay specification, given as a target, and a maximum value for the end-to-end transmission delay. Two classes of services are distiguished, *guaranteed* and *best-effort*. For guaranteed service, a resource reservation is made on the basis of the *linear bounded arrival process* abstraction; in this sense SRP offers hard guarantees for the expected quality of service.

A more general approach to hard guarantees is proposed by the Tenet Group [6]. The client specifies its throughput requirements in terms of a minimum and an average interpacket 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 a delay jitter upper bound (*bounded-jitter service*) [11]. A communication reliability requirement can also be specified with a probabilistic bound on packet losses.

A different solution is proposed in the Flow Protocol [13]: a traffic control algorithm is introduced to control the average transmission rate of data flows. The client can specify its traffic by means of an average transmission rate and an average interval, so that the network can reserve the resources required to satisfy the client's needs. The proposed algorithm has also a beneficial effect on the delay that packets will suffer in the network. However, no guarantee is given in terms of a bound on the communication delay. For these reasons, the Flow Protocol can be considered as offering hard guarantees only for what concerns traffic throughput.

The Stream Protocol Version II (ST-II) [4] requires that the client specify its traffic requirements with a set of throughput parameters such as, for example, the desired and the minimum packet size and rate. From these data and the current traffic load, the network evaluates the throughput and delay figures actually obtainable and proposes them to the client. Throughput is expressed by an allowed packet size and rate, while communication delay is represented by an accrued mean delay and delay variance. The capabilities of the network to guarantee the promised quality of service are difficult to determine, since neither in the protocol specification nor in its implementations (e.g. [9]) are algorithms presented for resource reservation and scheduling. However, the suggested solution of providing a guaranteed service only for the average data rate of each communication, and of sharing with the other connections additional network capacity for accommodating bursts, seems to be able to offer only soft guarantees.

In [3] a particular class of multimedia applications, called *playback* applications, is introduced. The clients of these applications can adapt to variations in the quality of service provided by the network and even endure temporary disruptions in the service. For this kind of application, a new type of real-time communication service is introduced in addition to the guaranteed one provided for non-playback applications. In this service, called *predicted*, the network attempts to deliver a service that satisfies the client's requests by reserving an amount of resources related to recent measurements of the traffic load; however, when the traffic conditions change, the quality of service provided will generally change as well.

The soft guarantees approach is likely to exhibit an important advantage when compared with solutions offering a guaranteed service: since all protocol computations and reservations are made on the basis of the current traffic situation, the amount of resources to be booked for each client in the network nodes may be smaller. This should allow the establishment of a greater number of communications and an improvement in the utilization of the network's resources. The latter effect could be due to the fact that clients of this kind of service will be able to endure reductions and disruptions in the service, as a consequence of the admission of new, more demanding, traffic or of unpredictable changes in the behavior of some other client.

The validity of the soft guarantees approach relies mainly on the existence of this very particular class of clients and multimedia applications. Indeed, as we have shown in the examples before, it is true that some applications have less restrictive requirements than others and show a more regular traffic pattern. However, this does not necessarily imply that such applications and their users will be willing to accept a service whose quality is not guaranteed in all conditions.

The level of flexibility available in most of the current real-time protocols is unsatisfactory, leading to inefficiencies in the network's utilization. The solution we propose for this problem is to introduce new mechanisms in the protocols that provide hard guarantees for quality of service, in order to improve their adaptability to changes in load conditions. To do so, we have devised a new, and as general as possible model for client-network interactions, and a new, architecture independent service interface that can be adopted by networks using different mechanisms to offer guaranteed quality of service. The introduction of the new model and service interface is advantageous also from the network clients' point of view since, by improving their capability to negotiate the quality of service with the network, it allows them to request a communication service that better conforms to their needs, and increases the chances to have such a service established.

3. The Client-Network Interaction Model

Most of the real-time protocols proposed in the literature include a very simple scheme for the interactions between the clients on one side, and the network on the other. A real-time communication client has in general a set of performance requirements reflecting the quality of service it expects to obtain from the network. In the case of protocols providing hard guarantees for the quality of service, the network can be considered as a *communication server* that, if sufficient resources are available, is able fully to satisfy the client's requirements for the whole duration of the connection.

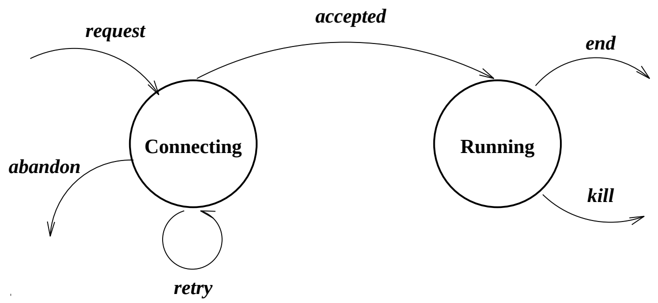

Fig. 1. States of a Real-Time Network Client

A client who requests the establishment of a real-time connection has to describes its requirements to the communication server. Then the server evaluates the request, and produces a positive or negative answer. In the first case, the server has been able to reserve the resources needed for establishing the connection and to ensure the requested quality of service; a positive answer enables the client to start the communication by sending packets along the established channel.

If the client receives a negative answer, it may abandon the request, or retry later with the same requirements, hoping that this time it will be accepted. The client may also decide to input a new request with different requirements, especially if additional information explaining the reason for the rejection was included with the network's answer. Once the connection has been established there is no other form of interaction between the client and the network, except that the client will eventually request the tear-down of the connection.

The behavior of a client of such a service can be described by means of a state diagram (Fig. 1), where only the states related to communication have been included. The client is in the *connecting* state when requesting a connection with a guaranteed QoS. If the server gives a positive answer (*accepted* transition) the client is allowed to move to the *running* state and to start the transmission. It will remain in this state until it closes the connection (*end* transition), or until killed for some reasons (e.g. a failure) by the network (*kill* transition).

The simplicity of this model can lead to inefficiencies in the network utilization and to a high number of rejected calls. In some cases, for example, the network could accommodate a request with slightly different performance or traffic parameters, but the client does not have enough information about the network's state to modify its requirements accordingly. In other cases, a saturation situation could be easily resolved by modifying existing connections.

Our intention in this paper is to examine the conditions for allowing a more complete dialog between the client and the network. We also want to see whether some parts of this dialog can be automated, and analyze the feasibility of its implementation.

4. A Profile for Real-Time Network Clients

We will develop our proposal by using as an example the Tenet real-time communication model. In the protocol suite based on the Tenet model [8], real-time communication is based on simplex fixed-route connections, called *real-time channels*, or simply *channels*. The establishment of a new channel is realized in four steps:

1. The client determines the values of the network parameters characterizing a connection to be established with a specific destination. The client is presumed not to have, in general, any information on the network's state.

2. The network receives the request and evaluates it. This evaluation is not centralized in a particular node, but is performed by submitting the request to each node on the path selected between the source and the destination. If a node has sufficient resources to accept the new connection without compromising the performance of the already existing channels, an adequate amount of resources is conditionally reserved for the new channel.

3. The request proceeds toward the destination until one of the following events occurs: (*i*) one of the nodes along the path cannot accommodate the new channel, (*ii*) the destination is successfully reached and can accomodate the new channel. In the first case, that node sneds a negative answer toward the source. In the second case, the destination sends a postive message back to the source.

4. Starting from the last node reached, the answer message proceeds backwards to the source, following the same path of the request message. If the answer is a rejection, the resources tentatively reserved are released; otherwise they are confirmed and the channel is established.

First, the client sends a request for a channel that will best fit its traffic and performance requirements. Then, the network replies with a *channel established* or a *channel rejected* message. In the latter case, a brief description of the cause of rejection is provided. In fact, the network only specifies the resource whose shortage was responsible for the rejection. This kind of information does not give much help to the client in deciding what to do, since the client has no quantitative data on the basis of which an acceptable request could be formulated. In some cases, the network might be very close to establishing the channel, while in others the current load conditions could preclude the acceptance of all practical requests.

If the connection has been refused, then the client has two possibilities. The first is to repeat the same request until it is accepted. In addition to generating a flood of establishment related traffic on the network, this solution will cause the client to wait and to waste some of its computing resources. Since rejections of channel establishments are most likely to occur during periods of heavy network load, this approach seems particularly inappropriate.

The second possibility is that the client modifies its request, if this is acceptable to its application, by changing the values of some of its requirements, and submits it again to the network. To do so, the client needs more detailed information about the current network state.

Furthermore, the network can profit from an improvement in the amount of information that a client provides when it requests the establishment of a connection. For example, a client could specify, as a performance index, a range of acceptable values rather than a single one. The network could then choose the value, compatible with the client's requirements, that fits best the current situation.

5. A Performance-Oriented Classification

As we have shown in the previous section, during its lifetime a real-time client is either in the connecting state or in the running state. In the Tenet scheme, to request the creation of a real-time connection a client has to specify the following parameters [5]:

- for the traffic description, the minimum packet interarrival time x_{min}, the average packet interarrival time x_{ave}, averaged over an interval I, and the maximum packet size s_{max};
- for the performance requirements, a bound on the end-to-end packet communication delay and, optionally, on its variation, or jitter;
- for the reliability requirements, a lower bound W on the probability of successful delivery of packets to the destination.

A real-time channel is called deterministic, statistical or jitter bounded depending on whether the delay bound is specified respectively *i)* as an absolute upper bound D, *ii)* as a bound D to be satisfied with client-requested probability Z, or *iii)* as a pair (D, J), where J is the client-requested delay jitter upper bound.

We now propose an extension to this classification, based on the capability of a client to be statisfied by a range of values for the traffic and performance parameters. A client is said to be *inflexible* when it has fixed requirements, i.e. requirements that do not provide the network with any latitude at the establishment of the connection. Because of this lack of latitude, for such a client the network's only possible answers to an establishment request are the rejection of the request or its acceptance as it is.

A client is said to be *flexible* when it is willing to accept a range of qualities of service. The client indicates its flexibility by specifying ranges of acceptable values for one or more components of the set of performance and traffic parameters. Note that a flexible client reduces to an inflexible one when all the ranges of its parameters go to zero.

Let us consider a client whose performance requirement is a bound D on the end-to-end packet delay. If the client is inflexible, only the value of this parameter needs to be assigned. If the client is flexible, it also has to specify a range of acceptable values for the specified quality-of-service index. In the case of the delay bound, this could be specified as a desired delay bound D and a delay range δ such that the network-proposed delay bound D_n is acceptable by the client if:

$$D \leq D_n \leq D + \delta$$

Since for this new class of clients the network can choose the deliverable QoS in a range of values, the probability of channel rejection is generally reduced without increasing the complexity of the channel establishment algorithm.

It might be argued that a network provider should adopt a conservative approach and always choose for a flexible client the value of each parameter most favorable to the network and least favorable to the client. Following such an approach, in the previous example the network would select a delay bound equal to the upper bound $D + \delta$.

Two considerations can be made against this argument. The first consideration is that client satisfaction, intended as a commitment by the network to adhere as much as possible to the desired (rather than the tolerable) QoS specification, should be privileged over other issues, especially when the pricing policy is appreciably influenced by parameter the parameter in question.

The second is that the amount and the kind of the resources to be allocated to a real-time channel are functions of all three client-specified quality of service parameters, i.e. traffic throughput, delay, and reliability. For example, in the case of a heavily loaded network where most of the communication delay is generally due to queuing, by choosing a lower delay bound the network could reduce the amount of buffers to be allocated to a channel in the nodes.

We would like to insist on the point that, in our proposal, even a flexible client would still receive a well-defined and guaranteed quality of service. Indeed, the flexibility of a client is related only to the establishment phase of a real-time connection. Once the connection is created and its QoS is determined, the corresponding values of the performance and reliability parameters are guaranteed for the whole duration of the communication.

6. A Time-Oriented Classification

In addition to the performance-oriented profile discussed in the previous section, we now propose another one based on the capability of a client to specify the expected starting time and duration of a real-time communication at the time such a service is requested to the service provider.

A duration parameter has never been considered in any of the real-time protocols proposed in the literature. However, we believe that the next generation of real-time network models should also cope with this aspect, since the ability to book a communication service in advance seems to be an essential requisite in several future multimedia applications scenarios.

In the current version of the Tenet protocol suite, the reservation of the resources for establishing a real-time connection is made just at the time at which this connection is needed. We propose an extended model where the client is given the possibility of booking in advance the resources it needs for its communication requirements. This should improve the network's utilization and reduce the probability that a request for one or more new channels is refused for lack of resources.

In the model we propose, a client can be defined to be *time-specified* or *time-unspecified*. In the first case, the client specifies to the network the time when the requested service should be made available, as well as its expected duration. This is done by using two new traffic parameters, i.e. T and ΔT, respectively. In the second case, the client does not specify any duration but only the desired starting time. A request for the immediate establishment of a connection will be specified by setting T equal to zero.

A reservation mechanism, to be successful, has to be enforced by means of control mechanisms. For this reason the network must verify that a time-specified client does not exceed the time interval granted to it, unless the amount of resources available in the network allows an extension of that interval without jeopardizing the satisfaction of the needs of other clients who are either running or have pre-reserved resources for the very near future.

Clients should be encouraged by the service provider by means of an appropriate pricing policy to specify a duration for their real-time connections. Knowledge of starting times and durations should allow the service provider to predict more accurately the evolution of the real-time load. Given the estimated duration of a connection, the network could decide to spend more time to establish an *optimized* connection for a long-duration client, or to devise, for a client demanding a large amount of resources, a path that can reduce the probability of saturating the capabilities of the network to accept real-time traffic.

The addition of two traffic parameters to the client-service interface may be seen as a new burden on the client. However, there are a number of multimedia applications where time constraints are already part of the problem. For example, teleconferences and collaborative work sessions are to be scheduled in advance like any traditional meeting, to allow participants to avoid conflicts with other commitments and to reserve the facilities needed.

7. Dynamic Management of Communications

We have outlined the main features of a new client-service interface that increases the flexibility of both parties during real-time channel establishment. However, it would be advantageous to exploit this greater flexibility also throughout the lifetime of a real-time channel. This feature would allow the network to increase its capabilities to accept real-time traffic, and should permit the introduction of load-balancing procedures and fault-tolerance mechanisms in the management of a network providing real-time communication services.

From the viewpoint of this dynamic management, the behavior of a client can be defined as being either *static* or *dynamic*. A static client requires that, once the connection has been established, the quality-of-service bounds will be obeyed during the whole duration of the connection. Thus, for a static client the QoS is constant throughout the session.

For a dynamic client, the status of the connection can be modified, with the client's agreement, even while transmission is taking place. A modification in the quality of an existing connection might be requested by the network (e.g. to react to the occurrence of particular traffic conditions, such as the saturation of resources by the real-time load) or by the client it (e.g., to adapt the channel to new conditions in the application).

In the first case, a dynamic client must have indicated at the time the channel is established, a range of acceptable values for one or more of the quality-of-service indexes.

In the second case, the network has to modify the allocation of resources to accommodate the new QoS requested by the client. Consequentely, the modification of a channel's parameters by the client is subjected to approval by the network, and, in general, can be treated as a request for a new connection.

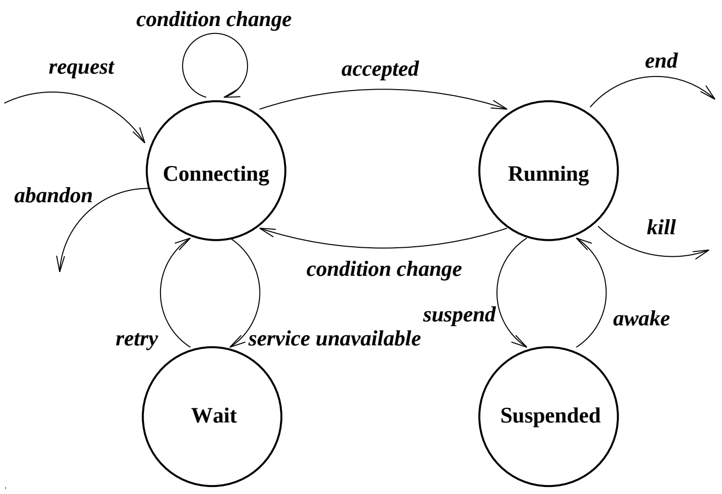

Fig. 2. The Extended State Diagram for Real-Time Network Clients

The behavior of a client can therefore be described by means of an improved state diagram (see Figure 2). In this diagram two more states have been added to that shown in Figure 1. A client is in the *wait* state whenever the service requested is still to be made available by the network. The *suspended* state includes all the situations when the network might have to suspend a running communication, as, for example, in the case a client's traffic would not conform to the original commitment, or for the occurrence of a failure in the network.

8. Conclusions

The problem of providing a flexible real-time communication service, suitable for a wide range of multimedia applications, can be solved by improving the flexibility of the hard guarantees approach to real-time service design.

In this paper we have presented a general model to improve the flexibility and the adaptivity of real-time communication protocols, and we have described how it can be applied to the Tenet real-time protocol suite.

The proposed model improves the utilization of network resources and increases the network's capability to face saturation of resources by real-time requests, while continuing to offer guaranteed qualities of service. A new profile for the clients of real-time communication services has been introduced, which takes into account the possibility of negotiating the QoS of a connection during its establishment, and of modifying the characteristics of an existing connection.

Acknowledgments

The authors are grateful to all members of the Tenet Group, and in particular to Amit Gupta, Jorg Liebeherr, Colin Parris, and Hui Zhang, for their comments and suggestions on this research.

References

[1] D. Anderson, R. Herrtwich, and C. Shaefer. SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet. Technical Report TR-90-006, International Computer Science Institute, February 1990.

[2] D. D. Clark, L. Chapin, V. Cerf, R. Braden, and R. Hobby. Towards the Future Internet Architecture. *Networking Working Group, Request for Comments*, N. 1287, December 1991.

[3] D. D. Clark, S. Schenker, and L. Zhang. Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism. In *Proceedings of SIGCOMM 92*, 1992.

[4] C. Topolcic ed. Experimental Internet Stream Protocol, Version 2 (ST-II). *Networking Working Group, Request for Comments*, N. 1190, October 1990.

[5] D. Ferrari. Real-Time Communication in an Internetwork. Technical Report TR-92-001, International Computer Science Institute, January 1992.

[6] D. Ferrari and D. Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE Journal on Selected Areas in Communications*, 8(3), April 1990.

[7] T. D. C. Little and A. Ghafoor. Network Considerations for Distributed Multimedia Object Composition and Communication. *IEEE Network Magazine*, November 1990.

[8] C. L. Lowery. Protocols for Providing Performance Guarantess in a Packet Switching Internet. Technical Report TR-91-002, International Computer Science Institute, January 1991.

[9] C. Partridge and S. Pink. An Implementation of the Revised Internet Stream Protocol (ST-2). In *Proceedings of Second International Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, November 1991.

[10] M. Stonebraker and J. Dozier. Large Capacity Object Servers to Support Global Change Research. Technical Report 91/1, Sequoia Technical Report, 1991.

[11] D. Verma, H. Zhang, and D. Ferrari. Delay Jitter Control for Real-Time Communication in a Packet Switching network. In *Proceedings of TriComm '91*, pages 35–43, Chapel Hill, April 1991.

[12] K. Watabe, K. Sakata, H. Fukuoka, and T. Ohmori. Distributed Conferencing System with Multiuser Multimedia Interface. *IEEE Journal on Selected Areas in Communications*, 9(4), May 1991.

[13] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet-Switched Networks. *ACM Trans. on Computer Systems*, 9(2), May 1991.