Network Support For Multimedia

A Discussion of the Tenet Approach*

Domenico Ferrari, Anindo Banerjea, and Hui Zhang
Tenet Group
Computer Science Division
University of California at Berkeley
&
The International Computer Science Institute,
Berkeley, CA

TR-92-072

November 1992

Abstract

Multimedia communication can be supported in an integrated-services network in the general framework of realtime communication. The Tenet Group has devised an approach that provides some initial solutions to the realtime communication problem. This paper attempts to identify the principles behind these solutions. We also describe a suite of protocols, and their implementations in several environments, that embody these principles, and work in progress that will lead towards more complete solutions.

*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.

1 Introduction

High speed networking has introduced 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. In the area of computer networking, multimedia applications can be considered a subset of the larger class of realtime applications, which include realtime distributed computations, remote control systems, and interactive applications. We shall therefore address the problem of realtime communication rather than the more restricted one of multimedia communication (see also §3.2). While circuit-switching networks can provide realtime support, the belief is now widespread that future integrated-services networks will use packet-switching techniques to achieve a more efficient usage of network resources. Since current packet-switching networks cannot guarantee the required levels of performance to their clients, the problem of how to support multimedia and other realtime applications in a packet-switching environment has become a pressing research issue.

The Tenet Group at the University of California at Berkeley and the International Computer Science Institute has been working since 1988 to provide a practical solution to the problem of realtime communication. The initial goal of the work was to devise and specify in all its details a set of algorithms that, when implemented in a network, would enable the network to offer a realtime service to its clients. Such a set of algorithms was called a *realtime communication scheme*, or a *scheme* for brevity. When the initial scheme (to be called Scheme 0 in this paper) was completed, the group was offered the opportunity to perform realtime communication experiments on the Xunet 2 wide-area networking testbed [FKK+92]. This, however, required several extensions to Scheme 0; the resulting scheme (Scheme 1 in the sequel) has been embodied in a suite (Suite 1) of realtime protocols designed by the Tenet Group.

The design of these schemes and suite required, as is the case with all designs, a number of compromises and restrictive assumptions. However, the group's previous publications paid much more attention to the features of the schemes and suite than to the distinction between the principles of the approach and the design compromises. This has unfortunately misled many discussions of our work. In this paper, we try to identify the principles that have guided, since the beginning, the Tenet Group's work, so that the debate can be focused on the crucial issues rather than on less important or unimportant considerations. These principles can be regarded as constituting and characterizing the *Tenet approach* to realtime communication. Figure 1 shows the relationships among the concepts mentioned in this section.

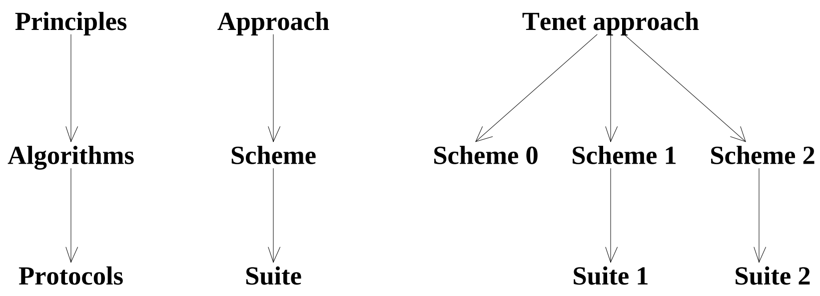

Figure 1: Components of the Tenet approach.

A large number of schemes can be constructed that are consistent with the Tenet approach. Some schemes may obey only a subset of the principles of the Tenet approach if the corresponding restrictions are deemed necessary to make the scheme's construction manageable.

We summarize the main features of Scheme 0 and Scheme 1 in Section 2 as a background for the following discussion. In Section 3, we first clarify the meanings of some of the terms we have used, and identify the most important compromises we made in our designs; then we discuss the principles that constitute the Tenet approach to realtime communication. Section 4 describes the design of Suite 1 and its relationships to both Scheme 1 and the networking testbeds for which it has been implemented. Some of the novel aspects of Scheme 2, now being designed by the Tenet Group, are outlined in Section 5. Section 6 briefly examines related work from the viewpoint of the Tenet principles. Section 7 concludes the paper.

2 The initial Tenet scheme and its extensions

2.1 The initial scheme

The initial Tenet scheme (Scheme 0) is described in detail in [Fer89], [FV90a], [FV90b], and [VZF91]. The network type it assumes has general topology, but consists of identical switches and identical hosts. Thus, Scheme 0 applies to a homogeneous network; internetworking is not supported.

The communication abstraction on which Scheme 0 is based is the *realtime channel*. For the purposes of Scheme 0 a *realtime channel* is defined as a simplex unicast end-to-end connection with performance guarantees and restrictions on traffic. A realtime channel is associated with a set of nodes and links within the network, through which the realtime data packets for the connection pass, and which we call the *route* of the channel. To satisfy the performance guarantees, we reserve resources in the nodes on the route, and perform admission control tests during the process of establishing the channel.

2.1.1 Service definition

A realtime communication service is defined by the interface it offers to its clients [Fer90]. The interface of a service based on Scheme 0 allows the clients to specify their loads and their performance needs, and can be thought of as leading to a contract between each client and the network.

The traffic specification to be given by the client includes parameters expressing the peak and average load on the network, and an indication of the burstiness of the load. These parameters, which are allowed to take values from a continuous range over the set of positive integers, are:

- Minimum packet inter-arrival time x_min
- Average packet inter-arrival time x_ave
- Averaging interval I
- Maximum packet size s_max

The performance parameters by which clients describe their requirements are:

- Delay bound D
- Delay violation probability bound Z
- Buffer overflow probability bound W
- Delay jitter bound J

Again, each of these parameters is allowed to take values from a continuous range. The delay violation probability bound (Z) allows the client to specify the hardness of the guarantee on the delay bound. For example, if Z is one, then the guarantee is *deterministic*, otherwise it is *statistical*. A guarantee on bandwidth is obtained from the traffic specifications, because the network promises to absorb the offered load as described by the client and provide delivery within the specified performance bounds. Note that J is a deterministic bound, and that its specification is optional.

The network checks its internal state and either accepts the client's request or rejects it with an indication of the reason for the rejection. If the network accepts the request, then this agreement takes the form of a contract between the network and the client that is binding on both. The client agrees to abide by the restrictions of the traffic specifications, and the network guarantees *a priori*, in the absence of network failures, to provide the specified performance levels as long as the client keeps its part of the contract.

2.1.2 The channel setup procedure

The mechanism used in Scheme 0 to set up a realtime channel follows fairly naturally from some of the basic principles of the Tenet approach, as we will show in Section 3.2. This mechanism can be summarized as follows. The network takes the client's traffic description and performance requirements. It chooses a route for the channel and maps the global requirements onto local requirements for each node on the route. In Scheme 0, to simplify our task, we rely on the routing algorithms existing in the network, even though this algorithm may ignore the special needs and constraints of realtime channels. The network then performs channel admission tests on the basis of the requirements, and if successful, makes reservations at each node. This is done in a distributed fashion during a round trip communication.

On the forward part of the channel establishment round trip, resources are reserved to get the best possible level of local performance. This implies giving the channel the lowest available local delay bound, and reserving sufficient bandwidth and sufficient buffers to reduce overflow probability to as close to zero as possible.

At the end of the forward pass, we have information about the end-to-end performance that can be achieved with the current level of *realtime load* (i.e., the resources already allocated to existing realtime channels) in the network. If an end-to-end performance bound obtained is better than the client's corresponding requirement, then the network can improve its efficiency by reducing the reservations made during the forward trip. This is done during the reverse trip, and the network keeps track of the performance level for each index obtained by the reduced reservations to make sure that it always satisfies the client's requirements even after this relaxation process.

The channel admission criteria and the reservation algorithms are based upon worst case analysis of the scheduling mechanisms (which, in Scheme 0, are assumed to be Multi-Class Earliest Due Date (MCEDD) in all hosts and switches) [FV90a, LL73] and the traffic specifications, so that after the setup process we can make *a priori* guarantees about the performance of the channel. The data packets for the channel always follow the route chosen during setup, so that resources are always available when data packets arrive at a node. Thus, the guarantees made by the network to the client are never violated.

2.2 Extensions to the initial scheme

As mentioned in Section 1, the characteristics of the Xunet 2 testbed were not in complete agreement with the simplifying assumptions on which Scheme 0 had been based. Xunet 2 is an internetwork, since its wide-area portion is an ATM backbone, connecting routers that are in turn connected to hosts through local-area FDDI rings [FKK+92]. Furthermore, while hosts and routers can run a realtime operating system whose scheduler can be programmed to obey the MCEDD discipline, the switches in the ATM backbone schedule cell transmission according to the Hierarchical Round Robin (HRR) discipline [KKK90], which is implemented in hardware and is therefore not easily modified.

The study of the extensions needed by Xunet 2 produced a channel establishment procedure that was quickly found to be applicable to a very broad class of internetworks. The procedure is a hierarchical one, and reflects the hierarchical structure of internetworks.

An establishment message is sent by the source host during the forward pass of the setup phase. The message is hierarchically structured and can be thought of as containing a frame for each level of the internetworking hierarchy that the message is currently in. As the message enters a subnetwork, a new frame is added to contain the traffic description and performance requirements for this level. As the message leaves the subnetwork and passes up to a higher level of the internetwork, the frame describing the subnetwork is popped off. A record describing the cumulative performance indices of this subnetwork (as if it were a simple link) is added to the frame that is exposed (corresponding to the next higher level of the hierarchy) [BM91].

The internetworks in which the scheme works include:

- subnetworks in which packet delays are (or can be) bounded by non-Tenet schemes;
- hosts, gateways, and switches in which packet transmissions are scheduled by different disciplines, even within the same subnetwork or at the same level in the internetworking hierarchy;
- subnetworks and subsubnetworks to any hierarchical depth.

A large number of scheduling disciplines are compatible with the existence of a realtime communication service in an internetwork. The conditions a discipline has to satisfy are quite broad; even First Come First Served can be used to provide such a service as long as some other simple provisions are made [Fer92]. The admission control tests and local bound computations that each node does will depend on its scheduling discipline [FV90a, BK92].

Scheme 1 also includes a new mechanism for jitter control, which had to be devised as one of the extensions of Scheme 0 since the mechanism used in that scheme does not apply to internetworks. The new mechanism can be used in the very broad class of internetworks to which the other extensions apply [Fer91].

3 The Tenet approach

Many of the characteristics of the Tenet schemes described in Section 2 were dictated by the desire to simplify their design and implementations. Which of their characteristics should be regarded as fundamental? Before providing an answer to this question in Section 3.2, we clarify in Section 3.1 the meanings of some of the terms we have used, and identify the most important compromises we made in our designs; the characteristics resulting from these compromises should therefore be regarded as properties of the schemes themselves rather than as fundamental principles.

3.1 Clarifications

Guaranteed performance does not necessarily mean deterministic guarantees. One of the important elements of the Tenet approach is the emphasis on guarantees. That is, once the network accepts a client's request for a certain level of performance, it should be able to guarantee it, at least in the absence of network failures, as long as the clients obey the constraints imposed by the traffic specifications. This rather strong emphasis on guarantees does not mean that we are only interested in *hard* or *deterministic* guarantees. Guarantees can also be *statistical* or *probabilistic*, and in fact this was the case even in the very first Tenet scheme (§2.1.1). It is true that the service provider, to prove that a statistical guarantee will never be violated, must allocate resources for the (statistical) worst case, and this is what Tenet schemes do. However, this allows the burstiness of client traffic to be exploited for statistical multiplexing; and thus less resources need to be reserved for each connection than would be needed to provide deterministic ones. It has been shown in [Ver91] that, by providing statistical guarantees instead of deterministic ones, the utilization of a link by realtime traffic can reach roughly 30% when the probability of a packet's missing its deadline is guaranteed to be under 10^{-7} and the burstiness (defined as the peak-to-average ratio) of the traffic is uniformly chosen between 2 and 20. The utilization can be raised to 65% when the probability of a packet missing its deadline is guaranteed to be under 10^{-2}. The utilization can be even higher with tolerant clients who do not object to larger fractions of packets missing their deadlines.

Having a general set of performance and traffic parameters does not necessarily lead to a complex client-service interface. In the Tenet schemes described in Section 2, the network provides a single parameterized abstraction to all realtime communication clients. One concern with this approach is that the client interface may be too complex because clients have to specify all the parameters each time they invoke the realtime service, and those parameters may not be easily estimated. It should be noted, however, that there are different levels of client-service interfaces in a network architecture. The general parameterized abstraction in the Tenet schemes is a service interface between the network layer and the transport layer. Additional library services can be added so that appropriate values of the parameters for certain common applications such as "low-quality video conferencing " can be pre-set in the system. Thus, most application programs need only to invoke the library routines, and can avoid specifying all the network layer parameters.

Client-service interactions need not be exceedingly simple. In Schemes 0 and 1, performance guarantees and traffic specifications are rigid: once a channel is established, its performance and traffic parameters cannot be modified. Also, when a channel setup request is rejected, the client is given a reason for the rejection, but no additional help from the service in making a decision about what to do is provided. There is no fundamental motive for this too simple interface, which was chosen as a design shortcut, and is going to be substantially improved in future schemes (see §5).

Routing of realtime channels need not rely on existing routing algorithms. Again, this was a decision made to reduce the complexity of scheme design, as we wanted to concentrate our efforts on the other algorithms. Indeed, we have always believed that the routing algorithm used by the channel setup procedure should explicitly take into account the channel's performance constraints and the realtime load in each node. This is another shortcut to be eliminated in future schemes (see §5).

A channel need not follow a single, fixed route. Single, fixed routes for channels are another simplification introduced in the design of Scheme 0 (and 1) for manageability. In future schemes we in intend to consider routes that are not simple paths; we could pick multiple disjoint paths [Max75] or single-failure-immune routes with redundant edges [ZS92] for reliability, or tree-shaped routes for multicast [Dee91]. We could also reroute a channel during its lifetime, as long as its current guarantees are not violated. All these options are under investigation currently.

Connection-oriented service does not necessarily mean long setup latencies. The Tenet schemes are connection-oriented and reservation-based. A typical objection to such an arrangement is that it may require a long setup latency. Notice that the reason the schemes are connection-oriented is that resources need to be reserved along the path to avoid congestion that would occur otherwise. A connectionless solution does not need to reserve resources; however, it needs to *probe* the network state to determine if the network is congested. For example, TCP uses a slow start strategy to avoid congestion, so that it takes a few round-trip times before the source can send at its full speed[Jac88]. Thus, in both cases, the connection-oriented Tenet schemes, and the connectionless TCP approach, the network needs to be *probed* before data can be sent at full speed. In the Tenet schemes, probing the network state actually results in reserving resources inside the network if they are available. It should not be surprising that both types of service need to probe the network state first; this is due to the fact that network resources are limited, and there is the possibility of congestion. Although it could still be argued that a connectionless service is more flexible in the sense that data can be sent at the same time as the network state is being probed, a similar technique can be used in a connection-oriented service. For example, by using the fast establishment technique described in [TV89], data can be transmitted at the same time the channel is set up.

3.2 The principles

In this section, we will outline the principles of the Tenet approach. These principles embody our view of some of the essential aspects of a good network/client interface, and more generally our view of what are the desirable characteristics of a solution to the problem of realtime communication.

Some of the basic principles of our approach are based upon a desire for generality, both in the interface and in the mechanisms employed by the solution. We do not want to create a new interface or a new protocol for each new type of multimedia or realtime traffic or for each new network type that we encounter. We want our schemes and protocols to be applicable to as wide an environment as possible. We realize that their incarnations as protocol implementations have to be specialized for the particular internetworks on which they are destined to run, but we want this specialization effort to be minimal, i.e., their portability to be maximae. Another fundamental belief we have is that many clients concerned about realtime performance require a service that is *predictable* and *dependable*. This section lists these beliefs and some of the essential features that follow from them, and attempts to justify both.

1) *The interface offered by the network should be general.* We would like to define an interface at the network level that satisfies as large a set of applications as possible. This leads us to the following essential consequences:

- *Continuous-media communication is a special case of realtime communication.* As already mentioned in Section 1, from the viewpoint of network performance requirements, continuous-media traffic is one of the types of realtime traffic. All the network-oriented requirements of continuous-media can be expressed in terms of bounds on quantitative performance indices, as the requirements of any other realtime application. Thus, the problem of continuous-media communication can be solved within the general framework of a realtime communication service. The Tenet approach is an approach to the design of such a service in a packet-switching environment. Since continuous-media networking is a special case of realtime networking, Tenet protocols are intended to guarantee a variety of quantitative performance bounds, not to deal separately with specific media. A Tenet suite does not include video protocols, voice protocols, movie protocols, and so on, but only the protocols that are needed to support the performance bounds required by the various realtime applications, including the continuous-media ones.
- *The network should provide a parameterized interface abstraction to all realtime applications.* A Tenet realtime communication service does not offer just a limited menu of possibilities (e.g., conference quality video, high definition television, CD-quality sound, animated images with voice track, and so on; or low-delay high-throughput communication, medium-delay low-throughput communication, high-delay low-jitter communication, and so on), but a continuum of values and combinations of values for the performance bounds. This does not prevent menus from being offered in those cases in which they are useful. For example, one such menu has been alluded to in Section 3.1 above when pre-stored performance bounds were mentioned. A Tenet service is, however, much more flexible, as it has the ability to map the requirements of all the applications onto one set of parameters, thus allowing menus to be extended, new menus to be created, and special out-of-menu requests to be easily handled.
- *Realtime communication requires several types of bounds to be guaranteed.* Throughput or bandwidth bounds are not sufficient to satisfy the requirements of realtime communication. While the number and identities of the elementary bounds in the minimum set have not been determined and cannot probably be decided because of our ignorance about future applications, we believe that such a set should also include delay bounds, delay jitter bounds, and reliability bounds, and that a realtime communication service should offer at least all of these bounds.

2) *The solution should be applicable to a wide variety of internetworking environments.* Most end-to-end communication will go through several networks. Realtime communication services that could not be easily implemented in a wide spectrum of internetworks would have limited value. Tenet schemes, with the only exception of Scheme 0, are easily portable to many different internetworking environments, as described in Section 2.2.

3) *The clients of a realtime communication service require predictable performance.* In our opinion, most of the clients concerned about realtime performance require mathematically provable, quantitative, though not necessarily rigid or deterministic guarantees. These clients will typically use a realtime service to create value-added services such as video or audio applications, which they will then offer to their users. To make any claims about the quality of service they can provide, they need to rely on a network service that guarantees performance.

We believe that the only way to provide predictable performance is to offer *a priori* guarantees to the client, and then ensure that these guarantees are not violated. However, network performance cannot be guaranteed in the face of unknown or unpredictable client behavior. Thus, the network/client interface model we propose takes the form of a contract between the network and the client, with the client promising to obey certain traffic restrictions and the network undertaking to provide a certain pre-agreed level of service (in the absence of network failures), if the client keeps its part of the contract. An essential aspect of a realtime communication contract is the pricing policy, which must be based on the principle of different prices for different services (e.g., different types of guarantees, different values of the performance bounds, and so on). If all services had the same or similar prices, the only incentive for clients to request the quality of service they really need rather than the best available one would be their desire to maximize the probability that their requests be accepted. This would, however, drastically reduce the level of resource sharing (i.e., the chances of admission), and financially penalize clients with lighter requirements if network costs had to be recovered.

This contract view of the network has some implications on the mechanisms of a Tenet scheme. The following three essential features are a direct consequence of this service model:

- *Admission control and resource reservation.* As the load on the network rises, a point is reached when to admit another channel would overload the system and prevent the performance requirements of this or existing channels from being satisfied. Every network offering performance guarantees must therefore test for this condition, and reject the new request at this point. To determine whether the guarantees will be violated after the addition of the new channel, we need to keep track of the resources the other existing channels may need for their realtime data transfers. Thus, resources have to be reserved for each admitted channel, but this is a packet-switching type of reservation, which allows resources to be used by other traffic at those times they are not needed by the channel for which they have been set aside. We believe in addition that after a new client request has passed all the channel admission tests, we should be able to offer *a priori* performance guarantees to the new client (and not violate any previous guarantees). An essential feature of the Tenet approach is the mathematical proof that, if the new request passes the admission tests, then the guarantees are met. This must be the case for all types of guarantees, including the statistical ones.

- *Connection-orientedness and pre-computed routes.* The network must control the amount of realtime load it needs to serve because, beyond a certain load level, performance guarantees cannot be met. It should also be able to give different qualities of service to data packets on different channels based upon the performance requirements of the clients. This calls for the introduction of intelligence within the internetwork (though not necessarily within all of its subnetworks), and the presence of state information that lasts for the duration of the channel's lifetime. In addition, the performance of a channel is closely tied to the route that the packets follow during transmission. A longer path generally has a larger delay associated with it. Also, the paths the realtime data might take need to have reserved resources on them, so that the local performances are good enough to meet their requirements. Thus, the possible paths must be known in advance if *a priori* guarantees are to be offered.

- *Protection of real-time channels.* Performance cannot be guaranteed if channels are not protected from misbehaving (realtime or non-realtime) clients. Realtime clients cannot be trusted to obey the traffic restrictions they themselves have specified. Either malice or failures may cause peak and/or average input rates to be larger than those taken into account when the channel was admitted. If the network had no way of isolating the misbehaving sources, the guarantees promised to realtime clients would be easily violated. To avoid this unacceptable situation, the network must either control the input rates explicitly on a per-channel basis or adopt scheduling algorithms that will do this automatically in the nodes (e.g., the rate-based service disciplines discussed in [ZK91]).

4 The Tenet realtime protocol suite

To be able to offer a realtime service to applications, a network must have suitable realtime features throughout its protocol hierarchy. Such a network should implement at least the following functions: (i) realtime data transfer at the internetwork layer, and (ii) end-to-end transport of realtime data. In addition, the network will have to deal with the tasks of (iii) channel setup and (iv) data transfer control. We chose to implement these four tasks as separate protocols in the first Tenet realtime protocol suite, Suite 1 (we did not design a suite based solely on Scheme 0; if we had, we would have called it Suite 0).

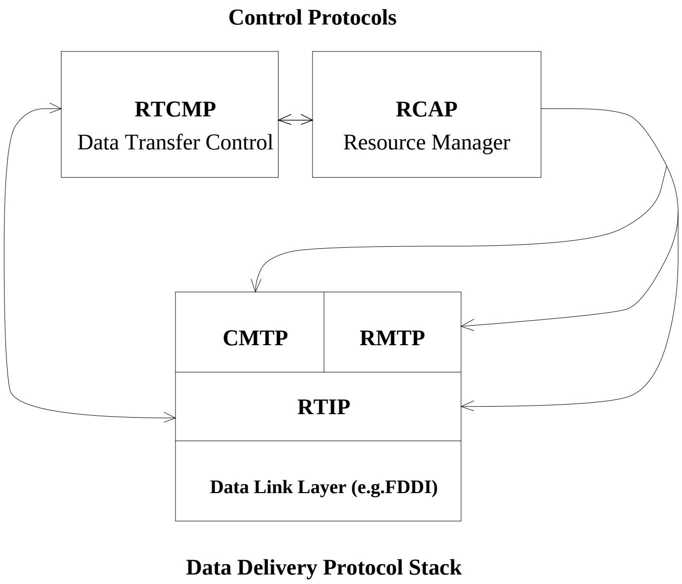

Figure 2: The prototype Tenet protocol suite (Suite 1)

The task of channel setup is performed by the Realtime Channel Administration Protocol (RCAP) [BM91]. RCAP is a control protocol that takes client requests containing traffic descriptions and performance requirements and sets up realtime channels in the internetwork. The data transfers at the internetwork layer are done by the RealTime Internet Protocol (RTIP) [ZVF92], which schedules data packets according to the resource reservations made by RCAP. At the transport level we designed two protocols: the Realtime Message Transport Protocol (RMTP) [ZVF92], which is intended for message-based realtime transport between endpoints, and the Continuous-Media Transport Protocol (CMTP) [WM91], which offers a stream-based interface and a time-driven mechanism for applications like video and audio.

The protocol that controls data transfers, primarily reacting to the detection of error conditions, is called the RealTime Control Message Protocol (RTCMP). RTCMP performs functions similar to those of ICMP in the DARPA Internet suite. All of these protocols have been designed to coexist with the DARPA Internet protocols, on which users are expected to rely for non-realtime communication in Xunet 2 and other environments.

At different times during the development of Suite 1, the opportunity arose to experiment with it in three additional testbeds:

- A simple local-area network assembled for the purpose for testing the Tenet protocols. It consists of several workstations connected by an FDDI ring.
- SequoiaNet, an internetwork connecting several University of California campuses and research institutes participating in Sequoia 2000, Digital Equipment Corporation's flagship project for the 1990's [Sto92]. This internetwork consists of local FDDI rings connected to T1 and T3 links via DECstations acting as routers. The primary application domain of SequoiaNet is scientific visualization for global change research, which involves storage, retrieval, and manipulation of the large data sets used in the earth sciences.
- Xunet 3, a higher speed (622 Mbps) topological subset of the Xunet 2 testbed involving HiPPI as well as ATM protocols at the data link layer. The main application domain of Xunet 3 is also scientific visualization, but with an emphasis in its Berkeley branch on medical imaging and the physics and engineering of materials.

None of these networks is more complicated from a Tenet scheme's viewpoint than Xunet 2, and some, for instance the local-area testbed, are substantially simpler. Thus, Scheme 1 and the implementation of Suite 1 were easily adapted to each additional testbed by suitably modifying the admission control tests and local bound computations. For obvious reasons, the first Suite 1 implementation completed was the one for the local-area testbed, and some of the initial measurements of its performance can be found in [ZF92].

5 The next scheme and protocol suite

In many important respects, Scheme 1 is still as simple-minded and primitive as Scheme 0. The basic communication abstraction is the simplex, unicast, single-path and fixed-path realtime channel. The performance guarantees a client can request and the traffic model a client must specify and obey are those presented in Section 2.1. The interactions between client and realtime service are the minimal ones mentioned in the same section. Routing is still assumed to be provided by at least one non-realtime protocol available in the internetwork (the exploitation of a non-realtime routing algorithm is greatly facilitated by the fact that none of the testbeds mentioned in Section 4 has a topology allowing for routing alternatives). Finally, there are only a few incomplete provisions for fault detection, and none for fault recovery.

The Tenet Group has recently begun investigating several of these areas, with the goal of determining what should be the features of a new Tenet scheme, to be called Scheme 2 in sequel. Once again, recognition of the limitations consciously imposed on the current scheme and suite does not imply that all of them will be removed or attenuated in the next scheme and suite. Since the group's resources are limited, and design, implementation, testing and experimentation are much easier when the number of new features is small, our current task is to select the few areas in which Scheme 2 will differ from Scheme 1, so that the development of Suite 2 will be manageable and at the same time result in a coherent, practically viable realtime service.

It is too early to specify, and even to predict, the characteristics and properties Scheme 2 will have. However, the importance of the collaborative applications of multimedia systems makes it likely that the new scheme will offer the multicast channel as its basic abstraction; such a channel, which trivially reduces to a unicast channel when only one destination needs to be reached, will probably still be simplex, but may have a (tree-shaped) route that can be modified during the channel's lifetime. If routes will be modifiable, Scheme 2 will not only have to include one or more (multicast) routing algorithms, but also be equipped with channel rerouting mechanisms. For highly reliable realtime applications, a channel's route may even consist of more than one path; these paths will have to be as disjoint as possible, and be either all in use or partially in use, partially in stand-by mode.

Another area in which some substantial improvements are almost certainly going to be made is the one of client-service interactions. After the rejection of a request (perhaps even before), a client needs to get guidance from the service about how to proceed so as to increase the probability of acceptance. Some clients require or prefer advance reservation of the service to make sure the service is available when needed. Traffic and performance specifications will probably not be augmented or modified, but the client model may be made more general by allowing these specifications to be more flexible, for example, introducing ranges for the bounds instead of forcing bounds to take on single values only; this would allow the service to choose each bound within its assigned range according to the realtime load existing on the network, and to modify it during the channel's lifetime if the need arises. If there is a mechanism for changing the parameter values of a live channel, this could also be used to accommodate client requests for such changes; these requests would be made by clients, for example, to reduce their charges when the requirements of their applications become less stringent.

Yet another area in which there ought to be a major difference between Scheme 2 and Scheme 1 is that of fault detection and recovery mechanisms. Predicting the reliability and fault tolerance features the new scheme will exhibit is even harder than for the areas discussed above, since the relevant work has started much more recently. However, we believe that Scheme 2 will be regarded as a major step forward in this area with respect to Scheme 1.

As for the architecture of Suite 2, the basic protocols will probably have the same tasks (and names) as those in Suite 1. However, some higher-level protocols, for example, for the management of multimedia conferences, might be added to them.

Of course, the features that will actually be incorporated into Scheme 2 and Suite 2 instead of being set aside for future schemes and/or suites will also be substantially influenced by the results of the experiments being performed on the implementations of Suite 1.

6 Other approaches

In this section, we briefly discuss the main differences existing, to the best of our knowledge, between the Tenet approach described in Section 3.2 and some of the other approaches to realtime communication. In the context of the DARPA Internet, the Stream Protocol Version II (ST-II) has been proposed to handle the requirements of realtime clients [Top90]. Most of the principles on which ST-II is based are in agreement with those of the Tenet approach, but the absence of the resource management algorithms (admission tests, local bound computations) in the scheme on which it is based makes it hard to decide whether several types of guaranteed bounds are intended to be offered to the clients, and whether a predictable quality of service can be expected. We believe it is possible to supplement the ST-II scheme with resource management algorithms that will satisfy these two principles, but ST-II also allows algorithms to be chosen that will not satisfy them. In other words, the ST-II approach seems to take an indifferent position with respect to the two principles in question.

The Session Reservation Protocol (SRP) is based on an approach similar to ours, though the scheme and the protocol design are quite different [AHS90]. The main divergence between the two schemes lies in their client interfaces; no statistical performance bounds (which can save considerable amounts of network resources, hence appreciably enhance the network's capacity) are offered to SRP's clients.

Our belief that several types of bounds are to be guaranteed is not shared by the Flow Protocol [Zha89], whose client-service interface allows a client to specify only an average throughput bound; the resulting delay bound can be computed, but not specified as an independent primary performance objective.

The Asynchronous Time Sharing (ATS) approach is based on a fixed menu of quality-of-service classes [HL91]. Each realtime connection can only obtain the performance guarantees corresponding to those of its class. This interface is not as general and flexible as that mandated by Principle 1 in the Tenet approach. An even more rigid interface is that of the Switched Multimegabit Data Service (SMDS), which is about to be offered by the telephone companies in several countries: the only guarantees offered by SMDS are that, within a Local Access Transport Area (LATA), the delay for at least 95% of the 53-byte cells will not exceed 130ms for T1 service, 20ms for T3 service, and 75ms for mixed T1 and T3 service [DKK90].

The predicted service proposed in [CSZ92] (unlike the "guaranteed service" mentioned in the same paper, which is based on deterministic bounds) does not offer *a priori* guarantees; thus, the quality of service that a client can obtain from it is not predictable. This is also the case of the approach described in [Flo92], whose differences from ours include its connectionless foundations and its discrete menu of service classes.

7 Conclusion

The Tenet Group at the University of California at Berkeley and the International Computer Science Institute has been doing research in the field of multimedia communication since 1988. The strategy followed by the group consists of designing a set of algorithms through which a network or a class of networks can support realtime (i.e., guaranteed-performance) communication; designing protocols that realize those algorithms; implementing those protocols for one or more specific networking environments; experimenting with these implementations under realistic multimedia applications; and, finally, using the results of the experiments to design the next set of algorithms, protocols, and so on.

At the basis of all of this work there has been, since the outset, a philosophy that in the group's previous publications was never clearly distinguished from the simplifications and compromises that had to be made to facilitate its initial incarnations. This paper has attempted to separate this philosophy, which we have called the Tenet approach, from the non-essential aspects of the initial Tenet schemes and of the first Tenet protocol suite. We hope that this clarification effort will lead to a better understanding of our work and its relationships to the work done by other research groups in an area that is still quite controversial, and in which the issues being debated are still clouded by a considerable amount of confusion.

8 Acknowledgements

Many individuals besides the authors have participated in the work discussed in this paper. Dinesh Verma made important contributions to Scheme 0 and its evaluation as well as the design of Suite 1, especially RTIP and RMTP. Amit Gupta, Carlyn Lowery, Bruce Mah, Mark Moran and Bernd Wolfinger contributed to the design of Suite 1 and of its protocols. Gupta and Mah also participated, and are still participating, in the implementations of the protocols; Francesco Majorana helped with that of CMTP. Tom Fisher designed and built the kernel support needed by the protocols. Amit Gupta, Jorg Liebeherr, Colin Paris, Jean Ramaekers, and Giorgio Ventre contributed to the initial investigations of some of the features being considered for Scheme 2. Riccardo Gusella, Srinivasan Keshav, Vern Paxson, Wolfgang Schroeder-Preikschat, and Clemens Szyperski provided useful comments on the manuscript.

References

[AHS90] 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.

[BK92] Anindo Banerjea and Srinivasan Keshav. Queueing delays in rate controlled networks. Technical Report TR-92-015, International Computer Science Institute, Berkeley, California, February 1992.

[BM91] 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, Springer-Verlag. Heidelberg, Germany, November 1991.

[CSZ92] 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.

[Dee91] Stephan E. Deering. *Multicast routing in a datagram internetwork.* PhD dissertation, Stanford University, December 1991.

[DKK90] F.R. Dix, M. Kelly, and R.W. Klessig. Access to a public switched multi-megabit data service offering. *Computer Communications Review*, 20(3):46–61, July 1990.

[Fer89] Domenico Ferrari. Real-time communication in packet switching wide-area networks. Technical Report TR-89-022, International Computer Science Institute, Berkeley, California, May 1989.

[Fer90] Domenico Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, November 1990.

[Fer91] Domenico Ferrari. Design and applications of a delay jitter control scheme for packet-switching internetworks. In *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 72–83, Springer-Verlag. Heidelberg, Germany, November 1991. Also in *Computer Communications* 15(6):367-373. July-August 1992.

[Fer92] Domenico Ferrari. Real-time communication in an internetwork. In *Journal of High Speed Networks*, 1(1):79-103,1992

[FKK+92] Alexander G. Fraser, Chuck R. Kalmanek, A.E. Kaplan, William T. Marshall, and R.C. Restrick. Xunet2: A nationwide testbed in high-speed networking. In *Proceedings of INFOCOM'92*, Firenze, Italy, May 1992.

[Flo92] Sally Floyd. Issues in flexible resource management for datagram networks. In *Proceedings of the 3rd Workshop on Very High Speed Networks*, Maryland, March 1992.

[FV90a] 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.

[FV90b] Domenico Ferrari and Dinesh Verma. Buffer allocation for realtime channels in a packet switching networks. Technical Report TR-90-022, International Computer Science Institute, Berkeley, California, June 1990.

[HL91] J. Hyman and A. Lazar. MARS: The Magnet II real-time scheduling algorithm. In *Proceedings of ACM SIGCOMM'91 Conference*, pages 285–293, Zurich, Switzerland, September 1991.

[Jac88] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM'88*, pages 314–329, August 1988.

[KKK90] Charles R. Kalmanek, Hemant Kanakia, and Srinivasan Keshav. Rate controlled servers for very high-speed networks. In *IEEE Global Telecommunications Conference*, pages 300.3.1 – 300.3.9, San Diego, California, December 1990.

[LL73] C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of ACM*, 20(1):46–61, January 1973.

[Max75] N.F. Maxemchuck. *Dispersity Routing in Store and Forward Networks*. PhD dissertation, University of Pennsylvania, May 1975.

[Sto92] Michael Stonebraker. An overview of the Sequoia 2000 project. In *Proceedings of CompCon 92*, San Francisco, CA, February 1992.

[Top90] Claudio Topolcic. Experimental internet stream protocol, version 2 (ST-II), October 1990. RFC 1190.

[TV89] P. Todorova and Dinesh Verma. Fast establishment of real-time channels. Technical Report TR-89-056, International Computer Science Institute, Berkeley, California, October 1989.

[Ver91] Dinesh Verma. *Guaranteed Performance Communication in High Speed Networks*. PhD dissertation, University of California at Berkeley, November 1991.

[VZF91] Dinesh Verma, Hui Zhang, and Domenico Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of Tricomm'91*, pages 35–46, Chapel Hill, North Carolina, April 1991.

[WM91] Bernd Wolfinger and Mark Moran. A continuous media data transport service and protocol for real-time communication in high speed networks. In *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 171–182, Springer-Verlag. Heidelberg, Germany, November 1991.

[ZK91] Hui Zhang and Srinivasan Keshav. Comparison of rate-based service disciplines. In *Proceedings of ACM SIGCOMM'91*, pages 113–122, Zurich, Switzerland, September 1991.

[ZVF92] Hui Zhang, Dinesh Verma, and Domenico Ferrari. Design and implementation of the real-time internet protocol. In *Proceedings of the IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems*, Tucson, Arizona, February 1992.

[ZF92] Hui Zhang and Tom Fisher. Preliminary measurement of RMTP/RTIP. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, CA, November 1992.

[Zha89] Lixia Zhang. *A New Architecture for Packet Switched Network Protocols*. PhD dissertation, Massachusetts Institute of Technology, July 1989.

[ZS92] Qin Zheng and Kang G. Shin. Fault-tolerant realtime communication in distributed computing systems. In *Digest of Papers, FTCS-22*, pages 86–93, Boston, MA, July 1992.