Efficient Group Communication with Guaranteed Quality of Service*

Clemens Szyperski   Giorgio Ventre†

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

ABSTRACT *A specific class of multimedia applications is expected to be of importance for future communication networks: Multi-Party Interactive Multimedia (MIM). Based on the isolation and characterization of MIM applications, concrete network support requirements are derived in this paper. The varying degree of connectivity, the vastly different sizes in terms of participants and the reliance on a guaranteed Quality of Service make MIM support a difficult problem. Starting with the definition of multimedia communication abstractions, principles of solutions are sketched. For an important subclass of applications a particularly efficient and practicable alternative implementation based on half-duplex channels is introduced. Finally, interfaces at both the transport and network layers are considered.*

1 Introduction

In the area of distributed systems, two events can be considered as among the major sources of the research activity on high-speed networking. The first one is the increasing demand in terms of communication bandwidth due to the efforts to make multimedia technology available in distributed environments. The second event is the presence of more stringent requirements in terms of performance and reliability of the communication system due to the isochronous nature of the new media involved. The concept of reliability itself is intended now in a broader sense, and includes the capability of a network to guarantee certain performance parameters for the whole duration of a communication.

Among multimedia applications, those based on multi-party interactive multimedia (MIM) [7], show an inherent complexity, stemming from the different media they use and the large number of participants they might involve. Due to their origin from interpersonal communication models, multi-party interactive applications have been, in general, rather vaguely defined.

Other proposals have been presented in the literature for the classification of MIM applications from the viewpoint of the support they require from the communication subsystem [1, 6, 8, 10, 11, 12, 13]. In this paper we propose a different approach to such classification based on the identification of a set of basic functional characteristics that this class of multimedia applications exhibits. The first goal of this research is to show that such characteristics can be effectively exploited to reduce the amount of resources required to provide real-time communication. The second goal is to still accomplish this result with a guaranteed quality of service according to the Tenet approach to real-time networking [2]. According to the Tenet approach means, first of all, support of a contract between network client and network provider; in this contract, the client specifies its traffic model and its Quality of Service (QoS) requirements, and the network, if it accepts, establishes a channel that guarantees the QoS as long as the client sticks to its traffic model. Secondly, for the network to be able to keep its contracts, it needs to protect well-behaved channels from misbehaving ones. Thirdly, the traffic model and the QoS requirements are to be specified in a general way, not pertaining to any specific class of traffic.

The solution we propose is based on a new abstraction, called the *Half-Duplex Real-Time Channel*. This solution reduces the complexity of the creation of network support for common MIM applications, and decreases the amount of resources to be reserved in the network.

We start, in section 2, from examples of multi-party applications to extract a set of attributes we believe to be the most significant for the support of these applications by means of computer-based tools. We focus our attention on those attributes related to the logical aspects of the information exchange, and make little assumptions on possible implementations in a real-time communication network.

In section 3 we broaden our scope by considering the size of a communication event in terms of the number of participants. Based on that, section 4 follows with basic approaches, and section 5 introduces the half-duplex real-time channel. Section 6 contains the required admission tests for the Tenet protocol suite, and section 7 proposes actual client service interfaces.

*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 sabbatical leave from Università degli Studi di Napoli "Federico II", Napoli, Italy.

2 Model of Interaction

Interpersonal communication models show differences in the way data is exchanged among involved participants. In order to offer computer-based multimedia tools implementing these and other models, we need to analyze how such differences influence the kind of communication service these tools require from the network. This can be done by considering, as the prevailing aspect, the *model of interaction* criterion, defined as the aspect that covers how interaction may occur among participants of distributed applications. For example, we are interested in applications based on multi-party interactive multimedia (MIM). Depending on the ability of a participant in such applications to produce and transmit multimedia data, we can identify three different kinds of interaction: *Dynamic*, *Static*, and *Controlled*.

We will present our conjectures by considering a computer supported multimedia environment where participants to a multi-party communication are geographically distributed entities (*nodes*), interacting through communication abstractions (*channels*) over a real-time internetwork.

A *dynamic* interaction is required for applications in which the amount of information that can be exchanged among all the participants is almost equivalent and the characteristics of the applications do not allow the determination of a particular subset of participants that will be the main source of the data flow. Since the coexistence of multiple flows of information should be, in these cases, permitted, support for simultaneous communications among the nodes should be provided by the network, in spite of any arbitration or control mechanisms that might be imposed by the users.

A *static* interaction is required in all applications where the information flows from a statically determined subset of nodes, or where the amount of data flowing from the other nodes does not justify the creation of a dedicated, real-time communication channel. Typical applications requesting such kind of interaction are those generally referred to as *telepresence* [5]. An example of this kind of application is *teleprogramming* [4], where a user receives a number of different media from remote devices, sensors or computers, typically to control or manipulate a remote apparatus, according to a many-to-one information flow.

A *controlled* interaction is required when the application specifically permits the identification of a dynamically variable subset of participants that contributes to the data flow. The subset may depend on the specific attributes of the participants. For example, in a conference, there exist subsets including the conference speakers and the conference managers respectively. Both sets are expected to contribute to the data flow more than the attendees. The identification of the subset of senders can also depend on application-specific rules and agreements.

Table 1 lists some interesting multi-party applications. The entries are meant to indicate the dominant cases, but should not be taken as discriminative in the mathematical sense.

Type	*Description*	*Model of Interaction*	*Data Flow*
Course Lecture	Long term setup, lecture with managed discussion	Controlled	1 —> N
Seminar	Special event version of Course Lecture	Controlled	1 —> N
Meeting	All participants actively send/receive data	Dynamic	N —> N
Public Hearing	Small subset of senders many receivers	Static	M —> N (M << N)
Conference Panel/Debate	Moderated Questions & Answers from a subset of participants	Controlled	M —> N (M << N)
CSCW	All participants actively send/receive data	Dynamic	N —> N
Telepresence	Many senders, one receiver	Static	N —> 1
Broadcasting Services	One sender, many receivers	Static	1 —> N
General Distributed Computing	All participants actively send/receive data	Dynamic	N —> N
Phased Distributed Computing	All participants actively send/receive data	Controlled	N —> N

Table 1: Characteristics of Selected MIM Applications.

3 Aspects of Size

So far we considered purely functional aspects of MIM applications and no limits of any sort were assumed for the capability of a real-time communication system to provide the services required by these applications.

We believe, however, that the important issue of size of MIM applications should be taken into account [10]. By *size* we refer to the expected bound on the number of participants. In particular, we are interested in how support from a real-time communication subsystem has to cope with this aspect.

Indeed, the maximum number of participants that a particular application may support depends on when it becomes infeasible either for the human user or for the service provider to add another participant. For example, from a logical viewpoint, the dynamic model of interaction of some MIM applications might require a fully interconnected communication structure. This requirement seems to dictate a small bound on the maximum number of participants, since the connection complexity grows quadratically, and the information to be consumed by each participant grows linearly with the number of participants. Therefore, the existence of such a bound is due not only to implementation reasons, as those related to the complexity of the required communication structure, but also to the limits inherent in the human capability of effectively perceive several simultaneous information flows.

We studied how size relates to the various applications introduced in a previous section (Table 1). The result is presented in Figure 1, where the number of participants and the model of interaction have been used as criteria.

From Figure 1 we observe that the upper bound on the number of participants is somehow related to the specific application. It should be noted that this figure is intended to be only indicative. Still, we can distinguish among small, large, and very large scale applications. As a first guess, we expect the bounds for each of the three classes to be around 20, around 100, and practically unbounded, respectively. Of course, more precise numbers require an analysis of individual applications.

4 Principles of a Solution

In this section we propose the principles for a solution of the problem of communication support for MIM applications. The first solution we propose is based on the simplex multicast connection, and will be presented in the framework of the Tenet real-time protocol suite [2].

We have individuated two main reasons for adopting such solution. The first one is that mechanisms for an efficient im-

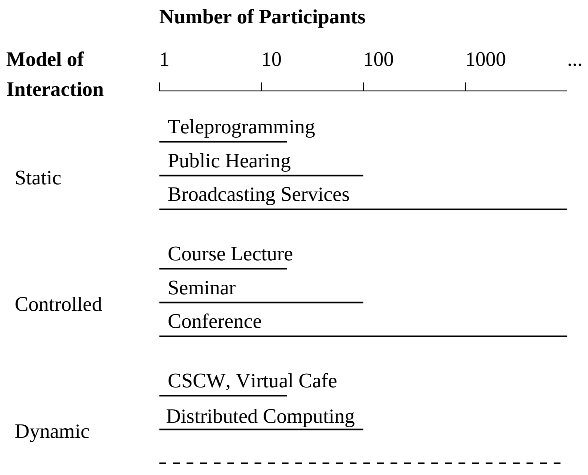

Figure 1: Size Ranges of MIM Applications.

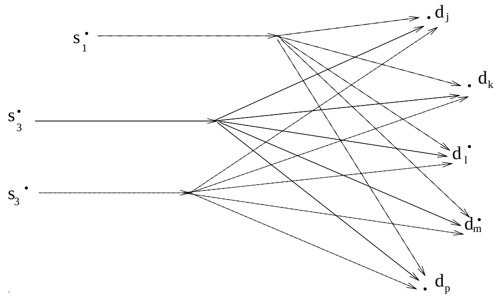

Figure 2: Possible Implementation of a Multimedia Channel.

plementation of multicast communication begin to be available in a number of packetswitching networks. The second is that a simplex multicast channel is a good compromise between the need of a mechanism more powerful than the simplex channel solution (on which the first Tenet scheme was originally based [3]), and the need of not to adopt mechanisms too powerful to be efficiently implemented.

An additional, and probably more important reason can be seen in the fact that the simplex multicast channel can be effectively used as a base for the realization of the communication services required by multimedia application. In fact, in the case of a static type of interaction, a simplex multicast channel seems to fit naturally in the scenario of a fixed separation of the application participants in sources and destinations, characterizing this kind of MIMs. For these applications the establishment of a real-time multicast connection from each source to all the destinations is sufficient, since the only form of data communication from the destinations to the source is generally related to application and communication control, and can be realized by means of non real-time data transmission. For example, in the case of an application requiring a static communication among a fixed set of information sources $\mathcal{S} = \{s_1 ... s_3\}$ and a fixed set of information sinks $\mathcal{D} = \{d_j ... d_p\}$, a solution with three real-time multicast connections can be devised, as shown in Fig. 2. In the case where interaction among the sources is also required, it is sufficient to implement the connections by considering the destination set $\mathcal{D}$ such that $\mathcal{S} \subset \mathcal{D}$.

When the level of interaction among the participants increases, then a richer communication structure has to be provided. However, the simplex multicast channel can still provide the basic component for an efficient implementation of such a structure. For example, in the case of a dynamic type of interaction, where all the application's participants are allowed to freely interact with the others, a complete network of real-time multicast channels connecting each participant with the others can be adopted.

The same solution, of course, can be applied in the case of applications characterized by a controlled type of interaction among the participants. Even though the above sketched solution appears to be straightforward and ease to implement, some considerations might be made regarding its efficiency. As we pointed in a previous section, a MIM can be defined as requiring a controlled interaction when the participants who can actively contribute to the data flow constitute a subset that can vary dynamically. Such subset can be usually identified by considering specific attributes of the application.

In the case of a teleconference, for example, all the speakers could simultaneously appear on the machines of the conference attendees, even though in practice only one among the speakers or a smaller subset of them is actively involved in the conference[1].

In the case of a computer supported lecture, the floor will be mainly taken by the professor; however, a good interaction between him and the students and also among the students themselves is highly desirable. For this reason, since in theory all the students could intervene during the lesson, it could be argued that a communication structure offering a real-time connection among all the participants is required. However the inherent order regulating the course of the lesson might be exploited to reduce the communication requirements.

Both the examples presented show that a more efficient solution for the implementation of the communication structure can be found, by considering the characteristics of the applications. In the teleconference example, the source set is usually small and only a subset of the speakers actively contributes to the conference at the same time; this characteristic should be exploited during the establishment of the channel, to determine some form of optimization in the reservation of the resources needed to guarantee a certain quality of service. In the case of applications where the source set is statically determined at the beginning of the application, we can adopt two different approaches.

5 An Alternative Solution

To present the first approach, we will consider again the lesson example. As we have seen, even though the professor is the only source of data for most of the time, all the students are supposed to intervene for a limited time during the course of the lesson. Again, a first solution to this problem could be to provide every participant with a multicast connection to all the others.

However, due to characteristics of the application, the multicast connections originating from the students would be mostly unused. A solution to this problem could be to implement a dynamic management of the connections, such that only the connection from the professor to the students

[1] Of course, some form of feedback from the conference attendees should be allowed, like questions or criticisms. In the case of large scale teleconferences, such feedback could be implemented by means of a non real-time communication channel, while for smaller scale MIM applications such as a teleseminar, a different solution could be feasible or appropriate.

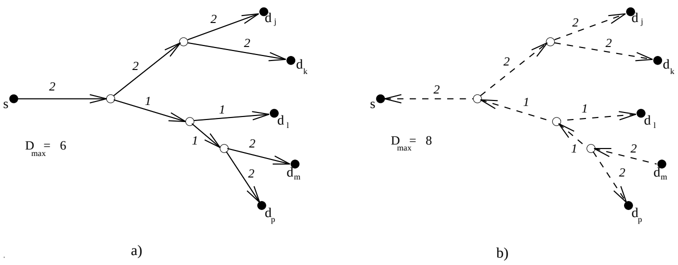

Figure 3: A Half-Duplex Multicast Connection.

is established at the beginning, while the other connections are created on demand, whenever a communication is requested by the students. This solution appears to be cost-effective, but it shows two major drawbacks: the first is that the time required to establish a new connection could be non-negligible, and, in general, unpredictable. The second is that the resources required for new connections on demand could be not promptly available.

The second solution is again based on an analysis of the application characteristics and on some ideas derived from the original Tenet scheme. In the lesson example, a Tenet-based approach would establish a group of simplex unicast channels with guaranteed QoS from the professor to the students. In an extended scheme with multicast connections capability the communication structure would appear as in Figure 3.a . Let us now suppose that in our application only one participant at a time can transmit data (i.e. is authorized to intervene). Since the sources are mutually exclusive, if the network could offer a guaranteed QoS in both directions of the same multicast connection, we could simply allow communication from the students by implementing, at the application level, control mechanisms realizing the required mutual exclusion. Figure 3.b shows the data flow when one of the students (d_m) starts to transmit data.

At the network layer, this control mechanism needs to be enforced, to avoid that an ill-behaving application could jeopardize the performance of other existing applications. In the Tenet suite, this enforcement can be done by means of the already existing mechanisms for rate control. With such mechanisms, whenever packets flowing from an application are exceeding the amount of resources originally allocated to it, thus violating the quality of service agreement with the network, they will be discarded with no harm to the other connections. Floor-control mechanisms within the network protocols can thus be avoided (e.g. [14]).

A new abstraction called the *half-duplex real-time channel* can now be introduced. The major problem of this introduction is that the reservation mechanisms in the Tenet scheme are dependent on the data direction over the channel. In the example of Fig. 3, all the network links have been assigned a delay bound in time units. When the data flows from the professor s to the students, the maximum delay the data will suffer is bounded by 6 units, while in the case where the data flows from d_m to all the other participants, the maximum delay will be 8 units. To avoid a too complex channel establishment mechanism, we believe that the specification of the required quality of service should be allowed only for the main direction of the data flow. In the lecture example, the quality of service requirements will be specified by the *main source* corresponding to the professor, while each one of the other possible sources (*alternative sources*), corresponding to the students, will only have from the network a resulting, yet guaranteed QoS.

The interconnection scheme proposed above can offer, in a real-time protocol suite, an efficient and cost-effective alternative to the already available, fully interconnected scheme.

Even though the former solution seems to restrict the capability of an application to select a real-time communication perfectly matching its needs, as instead allowed by the latter, we believe that the advantages characterizing a half-duplex multicast connection are well worth the limitations that might be associated to it. In fact, we believe it to be worthwhile to allow the explicit marking of a source as an auxiliary one, i.e. one that needs guaranteed QoS, but can live with most fixed QoS values as an outcome of establishment: If this information is made available to the network provider, significant gains in terms of establishment time and resource reservations can be expected. Thus, we consider the notion of auxiliary sources a concept rather than a side-effect of our Half-Duplex Channel proposal.

First, this solution greatly simplifies the establishment of a communication structure with guaranteed QoS, particularly for MIM applications characterized by a large number of participants (cf. Figure 1), since only the establishment of a multicast channel is required. This is particularly true in all cases in which a sort of asymmetry in the flowing of multimedia data among the participants can be observed; in such cases, the possibility of a reduced, but still guaranteed quality of service for less important connections is usually an acceptable drawback for a faster establishment of the complete communication structure.

Second, this solution provides savings in terms of resources required to establish a real-time connection. Following the traditional approach, the same application would have required the establishment of as many multicast channels as the number of participants, to allow each of them to have real-time interactions with all the others. In the next section we will focus on this last aspect and we will discuss the implementation of the proposed scheme within the Tenet protocol suite.

6 Implementation of the mechanism

To analyze the problems related to the implementation of a half-duplex multicast connection with guaranteed QoS, we will consider a simple example. In Fig 4.a we have a network node n_j with three channels passing through it. We now suppose that one of these channels, c_2, has to be half-duplex. In the Tenet approach this would imply to split channel c_2 in two different channels, $c_{2'}$ and $c_{2''}$ with similar characteristics, but flowing through node n_j in opposite directions (Fig. 4.b).

To determine the amount of resources required to accommodate both channels we have to perform first the local tests in all the nodes originally traversed by c_2 and then the final tests at the respective destinations. To understand if the mutual exclusion condition can be exploited to reduce the amount of resources required we will focus our attention to the local tests. The procedure for the establishment of a guaranteed real-time channel is described in detail in [3]. Here we will only remind that in each node three tests have to be performed: the *deterministic test*, the *statistical test*, and the *delay bound test.*

6.1 The Deterministic Test

According to the Tenet scheme, the traffic requirements of a connection are expressed in terms of the minimum inter-

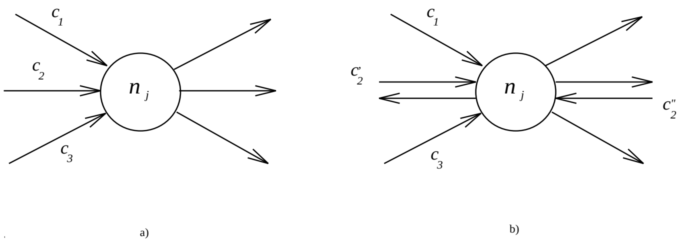

Figure 4: A Possible Solution for Half-Duplex Connections.

arrival times for the transmitted packets, x_{min}, the minimum value of the average packet inter-arrival time, x_{ave}, and the interval I on which these value have been computed. The quality of service required by the connection is expressed in terms of an absolute bound on the source-destination transmission delay, D_{max}. In this case the channel is called *deterministic*; if a deterministic channel j is established, in each node n on the channel path the packets will be subjected to a maximum delay $d_{j,n}$ computed at the end of the establishment phase. Alternatively, a requirement on the expected delay can be expressed in terms of a maximum value D_{max} to be satisfied by the packets with a probability greater than or equal to a value Z_{min} (*statistical* channel) [2].

For our simple example the deterministic test is :

$$\frac{t_1}{x_{min,1}} + \frac{t_{2'}}{x_{min,2'}} + \frac{t_{2''}}{x_{min,2''}} + \frac{t_3}{x_{min,3}} \leq 1 \tag{1}$$

where t_j is the maximum packet processing time for for channel j. If the real-time communication service requested has the same characteristics in both directions, channels $c_{2'}$ and $c_{2''}$ will contribute in the same way to the left term of the deterministic test. However, if the mutual exclusion is enforced they will never contribute at the same time. Indeed, since the deterministic test is based only on x_{min}, to have it passed is sufficient that both the new flows have the same value for this parameter of the original simplex channel. If the deterministic test has been passed by channel c_2, it will be passed also by channel $c_{2'}$ and $c_{2''}$. In the case that the flows in the two directions show different values for x_{min} and x_{ave}, then we have to verify the test for a channel c_2 such that $x_{min,2} = min(x_{min,2'}, x_{min,2''})$.

6.2 The Statistical Test

Given the mutual exclusion hypothesis, the two channels $c_{2'}$ and $c_{2''}$ can never be included in the same *overflow combination*. An overflow combination is a set of channels that, when simultaneously active for a sufficiently long time, may cause packets to violate the local delay bounds. Moreover, if we ensure that the number of packets received by node n_j over both channels $c_{2'}$ and $c_{2''}$ is, for each temporal window of duration I, equal to the number of packets that channel c_2 would have transmitted in the worst case in the same interval, we have exactly the same number of overflow combinations as if we had only channel c_2. To ensure this requirement, we have to enforce that the constraints for x_{min} and x_{ave} will be respected even in the worst case.

The constraint over x_{min} is given by that of the deterministic test. Regarding the condition over x_{ave}, we have two possibilities. The first one is to impose that whenever there is a change in the transmission direction (i.e. $c_{2'}$ stops while $c_{2''}$ starts to transmit, or vice-versa), the condition is respected in the worst case. Such case is verified when, immediately before stopping, channel $c_{2'}$ transmits at the maximum possible rate $x_{min,2'}$ for the longest possible interval $I_{min,2'} = I_{2'}\frac{x_{min,2'}}{x_{ave,2'}}$, and immediately after starting, channel $c_{2''}$ transmits at the maximum possible rate $x_{min,2''}$ for the longest possible interval $I_{min,2''} = I_{2''}\frac{x_{min,2''}}{x_{ave,2''}}$.

To avoid this problem, we have to impose a time interval δ between stopping the transmission in one direction and starting to transmit in the opposite direction. In the case of a switch from $c_{2'}$ to $c_{2''}$, we can determine such interval by imposing the following condition:

$$\delta_{2'} + I_{min,2'} \geq max(I_{2'}, I_{2''}) \tag{2}$$

In the case that $I_{2'} = I_{2''} = I_2$, from the above condition we have that the time interval should be:

$$\delta_{2'} \geq I_2 - I_{min,2'} = I_2(1 - \frac{x_{min,2'}}{x_{ave,2'}}) \tag{3}$$

Similarly, for a switch from $c_{2''}$ to $c_{2'}$, we have to impose the following time interval:

$$\delta_{2''} \geq I_2 - I_{min,2''} = I_2(1 - \frac{x_{min,2''}}{x_{ave,2''}}) \tag{4}$$

To avoid to have to impose this time interval before switching from one transmission direction to the other, a second solution is possible. It consists of over-dimensioning the original channel, (in our example c_2) in order to have that it could endure the worst case situation we depicted above. In this case we have to properly choose I_2 and $x_{ave,2}$ such that:

$$\frac{I_2}{x_{ave,2}} = (\frac{I_{min,2'}}{x_{min,2'}} + \frac{I_{min,2''}}{x_{min,2''}}) \tag{5}$$

and from the definition of I_{min},

$$\frac{I_2}{x_{ave,2}} = (\frac{I_{2'}}{x_{ave,2'}} + \frac{I_{2'}}{x_{ave,2''}}) \tag{6}$$

In the hypothesis that $I_{2'} = I_{2''} = I_2$, we have that:

$$x_{ave,2} = \frac{x_{ave,2'} \cdot x_{ave,2''}}{x_{ave,2'} + x_{ave,2''}} \tag{7}$$

And indeed, if we suppose that $x_{ave,2'} = x_{ave,2''}$ we have exactly the result we would expect:

$$x_{ave,2} = \frac{x_{ave,2'}}{2} \tag{8}$$

The advantage of this last solution is that in this case there is no need for a transition interval during the direction switch over the channel. Since the imposition of this interval should be done at the application level, the last solution seems to be simpler to implement, for it does not require any additional information flow from the protocols to the application. The drawback, of course, is that we now have to reserve more resources than by adopting the former solution.

[2]For the sake of brevity, a third type of real-time service, called jitter controlled, will not be considered.

6.3 The Delay Bound Test

The delay bound test determines the minimum delay bound to be assigned to the channel being established so that scheduler saturation can be avoided in the node. To do so, the test takes into account the service time of packets flowing through the node. This depends on the local processing time plus the data link processing delays. The former can be shared for $c_{2'}$ and $c_{2''}$ under the mutual exclusion assumption, while the latter is based on two different outgoing links that individually require protection in the worst case.

Hence, the delay bound test requires checking for the sum of the data link processing delays. In simple words, the delay bound test ends up in doubling the resources allocated in terms of overall use of bandwidth. This is to be expected since the two alternate flow directions present at each edge of a half-duplex tree typically map to different links, both requiring reservations.

To conclude, we can say that the costs of establishing a half-duplex multicast tree are at most twice that of a simplex multicast tree, independently of the number of participants. In the case where the processing nodes are the bottleneck (i.e. bandwidth allocation is not expected to be a limiting factor), and where the optimizations described above are applied, the costs for a half-duplex tree can even approach that of a simplex tree.

7 Towards Interfaces for the Control Protocols

Based on our analysis of the application requirements and our investigation of various mechanisms to support these, we now propose an interface for the control-related, applications-specific *transport* layer. The intent is to provide a building block that allows *session* or higher layers to compose more powerful communication abstractions. We restrict ourselves to the characteristic parameters of the provided abstraction. The fundamental idea is that for a channel there is a common traffic specification plus QoS requirements per destination. This is seen as a *contract*: If a channel client stays within the traffic specification, then the channel provider is obligated to guarantee the required QoS, or to refuse establishment of the channel. If an application asks for high QoS it increase the likelihood of refusal. A possible workaround is the introduction of application/network negotiations [9]. A different possibility, and a prefered one for its avoidance of unpredictable negotiation cycles, is the specification of QoS requirements by means of ranges where the desired value and the maximally tolerable deviation are given. For example, specifying a delay bound by the range $(D_{max}, \Delta D_{max})$ asks for a desired delay bound D_{max}, but regards all delay bounds up to $D_{max} + \Delta D_{max}$ as acceptable. This is indicated using the notation "delay-bound $D \in [D_{max}, D_{max} + \Delta D_{max}]$" [3]. The traffic and performance characterizing parameters have been introduced in [3].

7.1 Transport Layer

The transport layer provides the control abstraction for *single-media logical channels with QoS*. It computes the routes and the resource requirements. The resource requirements include necessary buffer space as well as parameters such as locally permissible scheduling delays. In the following specification, the curly brackets indicate one or more instances of the parenthesized definition.

T-Channel = (*Traffic-Spec*, *Src-Spec*, *Dst-Spec*, *Schedule-Req*, *Indications*)
  Traffic-Spec = (*Max-Packet-Size*, *Min-Interarr-Time*, *Avg-Interarr-Time*, *Avg-Interval*)

  Src-Spec = (*Main-Src*, {*Alt-Src*})
    Main-Src = *Alt-Src* = *src-id* (logical node id)

  Dst-Spec = {*Dst*}
    Dst = (*dst-id*, *Performance-Req*, *Fault-Tolerance-Req*)
      dst-id logical node id
      Performance-Req = (*Delay-Bound*, *Jitter-Bound*, *Arrival-Prob*, *Timeliness-Prob*)
        Delay-Bound = $(D_{max}, \Delta D_{max})$
          $\rightsquigarrow$ delay bound (in seconds)
          $D \in [D_{max}, D_{max} + \Delta D_{max}]$
        Jitter-Bound = $(J_{max}, \Delta J_{max})$
          $\rightsquigarrow$ jitter bound (in seconds)
          $J \in [J_{max}, J_{max} + \Delta J_{max}]$
        Timeliness-Prob = $(Z_{min}, \Delta Z_{min})$
          $\rightsquigarrow$ probability of keeping delay bound
          $Z \in [Z_{min} - \Delta Z_{min}, Z_{min}]$
        Arrival-Prob = $(W_{min}, \Delta W_{min})$
          $\rightsquigarrow$ probability of packet arrival
          $W \in [W_{min} - \Delta W_{min}, W_{min}]$

      Fault-Tolerance-Req = ...
  Schedule-Req = (*Absolute-Bounds*, *Span-Bound*)
    Absolute-Bounds = (T_{min}, T_{max})
      $\rightsquigarrow$ opening/closing times (in seconds after some T_0)
      $T_{min} \leq T_{open} < T_{close} \leq T_{max}$
    Span-Bound = $(S_{min}, \Delta S_{min})$
      $\rightsquigarrow$ duration span (in seconds)
      $S \in [S_{min} - \Delta S_{min}, S_{min}]$

  Indications = (*Sharing-Indication*, *Priority-Indication*)
    Sharing-Indication = *sharing-id*
    Priority-Indication = *prio-level*

Some of the parameters require additional rationales beyond what has been argued in the previous sections.

- *Src-spec*: The lifetime of a transport channel is equivalent to the time of membership of the channel's main source, i.e. the main source cannot dynamically leave and re-join its channel. According to the half-duplex model, a set of alternative sources can also be specified.
- *Advance reservation interval*: In a scheme asking for admittance control based on available resources it is important that advance reservation can be made in order to enable reliable planning of events. We propose to specify an advance reservation interval as a channel attribute consisting of four parameters (grouped into *Schedule_Req*). This attribute pins the opening time T_{open} and closing time T_{close} by giving absolute lower and upper bounds (T_{min}, T_{max}), and by specifying the desired duration of the session as a range $(S_{min}, \Delta S_{min})$. The reservation scheme is expected to establish the maximum possible subset of the requested interval. Considering a set $\{n_1, \ldots n_N\}$ of nodes n_i that have to reserve resources for this interval. Each node may well have a different subset interval that it could reserve. The result will be the intersection of all these, arbitrarily reduced

[3] Marginal values are considered to be inclusive.

to S_{min} if necessary:
$[T_{open}, T_{close}] = \bigcap_{i=1}^{N} [t_{open,i}, t_{close,i}] \subseteq [T_{min}, T_{max}]$, and $S_{min} - \Delta S_{min} \leq T_{close} - T_{open} \leq S_{min}$.

- *Fault Tolerance*: The support of fault-tolerance in the domain of interactive real-time applications is an open research problem. However, we consider it important and thus mention how we expect fault-tolerance requirements to be specified, i.e. as destination-specific QoS parameters.

- *Indications*: The *sharing-id* enables sharing of resources allocated by the transport layer (buffer space, bandwidth, processing time) for the handling of packets of channels that (above transport) are known to be mutually exclusive: Two channels with the same *sharing-id* are by contract with higher layers guaranteed to be mutually exclusive. The *prio-level* expresses a general notion of priority to be exploited to decide conflicts during channel establishment or re-establishment. Both indication parameters could be extended to convey more information. For example, the *sharing-id* could be replace by a pair (*sharing-id*, *sharing-traffic*) to allow modelling situations where for example two sources are allowed to send simultaneously.

7.2 Network Layer

The network layer provides and uses resources as computed by the transport layer and controls local routing of packets. To do so, the routing tables are created and distributed to all nodes seen by the network layer. Also, the network layer implements the packet queueing and scheduling mechanism. For this purpose, the appropriate information needs to be available, e.g. the local processing deadlines and the maximum number of buffer-space to be used by a channel. Note that the network layer implements rate controlling by making sure that a channel does not exceed its assigned resources and by dropping or delaying packets otherwise. This is important to ensure proper protection of other real- and non-real-time channels from a malfunctioning real-time channel client [3].

8 Conclusions

MIM applications are very demanding in terms of network support. We have shown that some characteristics of such applications can be effectively exploited to reduce the amount of resources required to provide quality-of-service guarantees in communication.

To allow this exploitation, we have introduced a new communication abstraction, called the half-duplex real-time channel. We have also shown how this abstraction can be efficiently implemented over the support for simplex multicast connections characterizing recent proposals for communication networks. The proposed solution reduces the complexity of the establishment of proper network support for applications with certain communications patterns, and greatly decreases the amount of resources to be allocated. These results have been accomplished in the framework of the Tenet approach to real-time communication, i.e. still providing guaranteed quality of service.

Future work will include the improvement of the algorithms for the establishment of a half-duplex real-time channel. In particular, we are interested in understanding to what extend we could allow quality of service requirements for auxiliary sources attached to a half-duplex channel, and what mechanisms would be needed to support this.

Acknowledgments

The authors are especially grateful to Domenico Ferrari, Amit Gupta, Bruce Mah, Mark Moran, Colin Parris, Makiko Yoshida, and Hui Zhang, for their ideas, comments, and criticisms. Wendy Heffner and Ginger Ogle are acknowledged for their contribution to an earlier version of the MIM applications taxonomy.

References

[1] Yee-Hsiang Chang. Remote Conferencing: Statement of Desired Functions. *Internet Engineering Task Force, Remote Conferencing BOF Draft*, October 1992.

[2] D. Ferrari, A. Banerjea, and H. Zhang. Network Support for Multimedia: a Discussion of the Tenet Approach. Technical Report TR-92-072, International Computer Science Institute, November 1992.

[3] 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.

[4] J. Funda. Teleprogramming: Towards Delay-Invariant Remote Manipulation. Technical Report MS-CIS-91-40, University of Pennsylvania, Dept. of Computer and Information Science, May 1991.

[5] D.R. Hofstadter and D.C. Dennett. *The Mind's I: Fantasies and Reflections on Self and Soul.* Basic Books, New York, 1981.

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

[7] M. Moran and R. Gusella. System Support for Efficient Dynamically-Configurable Multi-Party Interactive Multimedia Applications. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, November 1992.

[8] J.C. Pasquale et al. The Multimedia Multicast Channel. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, November 1992.

[9] J. Ramaekers and G. Ventre. Client-Network Interaction in a Real-Time Communication Environment. In *Proceedings of IEEE GLOBECOMM '92*, 1992.

[10] E.M. Schooler. The Impact of Scaling on a Multimedia Communication Architecture. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, November 1992.

[11] H. Schulzrinne. A Transport Protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications. *Internet Engineering Task Force, Internet-Draft*, October 1992.

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

[13] R. Yavatkar. Issues of Coordination and Temporal Synchronization in Multimedia Communication. In *Proceedings of IEEE Multimedia '92*, Monterey, 1992.

[14] R. Yavatkar. MCP: A Protocol for Coordination and Temporal Synchronization in Multimedia Collaborative Applications. In *Proceedings of 12th IEEE International Conference on Distributed Computer Systems*, June 1992.

All publications of the Tenet group are available via anonymous ftp from tenet.berkeley.edu, directory pub/tenet; an index is in file 00INDEX.