Efficient Multicasting for Interactive Multimedia Applications

Clemens Szyperski* and Giorgio Ventre*†

TR-93-017

March 1993

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.

*The Tenet Group, Computer Science Division, Department of EECS, University of California, Berkeley and International Computer Science Institute, Berkeley. E-mail: {szyperski,ventre}@icsi.berkeley.edu

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

1 Introduction

Multimedia applications, in particular those based on multi-party interactive multimedia (MIM) [10], 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. To avoid confusion, we included in Appendix A a list of terms that will be extensively used in the rest of the paper.

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 [3, 9, 11, 12, 13, 16, 17]. 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 [5].

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 from examples in the area of interpersonal communication (section 2) to extract a set of attributes we believe to be the most significant for the support of these examples 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.

Section 3 follows with an extension of the classification to computer-based applications. In section 4 we broaden our scope by considering the size of a communication event in terms of the number of participants. The requirements of MIM application clients are then stated in section 5, where we first refer to the important concept of *Quality of Service* (*QoS*). Thereafter, channel abstractions are introduced in section 6 to aid in structuring the requirements. Based on that, section 7 follows with basic approaches, and section 8 introduces the half-duplex real-time channel. Section 9 contains the required admission tests for the Tenet protocol suite, and section 10 proposes actual client service interfaces.

2 Functional Criteria of MIM Applications

We now propose a classification for some typical examples of social and professional communication and interaction, based on attributes primarily related to the functional aspects of the information exchange:

- *Model of Interaction*: This aspect covers to how interaction may occur among participants. Depending on the ability of a participant to produce and transmit multimedia data, we can identify three different kinds of interaction:

  Dynamic: All participants are allowed to freely send data without any form of restriction, even though some sort of control is usually implicit for this model of interaction. For example, social (e.g. parliamentary) rules or simply good manners apply to applications such as meetings. However, in this model we assume that no control is enforced by the application. A meeting of executives, a project design session, or an informal gathering of people are typical cases that may be included in this category.

  Static: This is the case where the participants that are allowed to send belong to a statically determined subset that will not change throughout the application's lifetime. As an example, we can consider a public hearing or a musical performance, where only a small number of the participants (i.e. the official speakers and the music players respectively) produce information, while all other attendees can only receive.

  Controlled: The subset of participants allowed to send data can change during the evolution of the application, but such a change is subject to the occurrence of conditions or to rules depending on the application itself. In this category fall examples like course lectures, seminars, or conferences, where the role of speaker can temporarily be assumed, in principle, by all the participants, as is the case if a lecturer occasionally allows the students to ask questions.

- *Data Flow*: This characteristic is related to how the information flows among the participants. In some cases, a one-to-many data-flow can be observed, while others are characterized by a more symmetric, all-to-all type of flow. An example of the first type is a seminar, where both the speaker and the attendees are expected to intervene only one at a time, while an example of the second case can be found in meetings where multiple communications can freely and informally coexist. In other models of interpersonal communication, like panels and debates, a $\mathrm{M} \rightarrow \mathrm{N}$ data flow can be observed, where usually $\mathrm{M} \ll \mathrm{N}$.

- *Accessibility*: In some cases, the access to the information flow is restricted, and some sort of control is enforced for billing or security reasons. This implies that the participants must undergo an admission or monitoring process.

Controlled: A course lecture is a typical example of a communication model where access is *controlled*. In fact, to be allowed to attend a lecture, a participant should be enrolled in that class. In other cases, such as business meetings, access is restricted for obvious reasons.

Uncontrolled: For some models, no kind of control need to be enforced over the participants. In a public hearing, as in the case of a City Council audition, everyone is usually permitted (and encouraged) to attend. There are also examples, as in a seminar-like communication, where access rules are imposed on a case by case basis.

Access control can be enforced by the existence of participant lists. Sometimes a participant list is prepared beforehand. In other cases, this list is dynamic, and may be modified during the event. However, for a large class of applications a hybrid situation is expected, where a certain subset of the participants is known when the communication is planned (*bounded subset*), while it is anticipated that the people originally not included in the list join it later. An example is a conference where a number of attendees announced their participation in advance, while the others have joined it only at the last moment. The two groups differ in the way they are treated. Participants who are known in advance are guaranteed to be accepted, while the others have to rely on the residual availability of resources (i.e. seats).

- *Event Scheduling*: There are a number of examples where time limits are essential. For example, conferences and collaborative work sessions are to be scheduled in advance, to allow participants to avoid conflicts with other commitments and to reserve the facilities needed. Therefore, we believe that the following two categories can be distinguished:

  Planned: This is the case when the communication is regularly scheduled, or more generally, just scheduled in advance. In addition to allow participants to schedule their attendance in advance, the availability of information like starting time and expected duration may be used for advance reservation of facilities (e.g. a room).

  Unplanned: In some cases, nothing is known about when the communication will occur. A typical example is a casual encounter of coworkers that develops into a discussion over work-related matters, or any meeting that is organized impromptu. Of course, the possibility of successfully having a communication of this kind is subject to the current availability of needed resources.

Type	Description	Model of Interaction	Data Flow	Accessibility	Event Scheduling
Course Lecture	Long term setup, lecture w/ managed discussion	Controlled	1 -> N	Controlled	Planned
Seminar	Special event version of Course Lecture	Controlled	1 -> N	Controlled Uncontrolled	Planned
Casual Meeting	All participants actively send/receive data	Dynamic	N -> N	Uncontrolled	Unplanned
Business Meeting	All participants actively send/receive data	Dynamic	N -> N	Controlled	Planned
Public Hearing	Small subset of senders many receivers	Static	M -> N (M << N)	Uncontrolled	Planned
Conference Panel/Debate	Moderated Q&A from a subset of participants	Controlled	M -> N (M << N)	Controlled	Planned

Table 1: Functional Characteristics of Interpersonal Communication Models.

Table 1 summarizes what we have presented in this section. It shows how a number of typical models of interpersonal communication fit into the classification proposed above. It should be noted that the table entries are meant to describe the dominant cases, and should not be considered as discriminative in the mathematical sense.

Even though other important examples have probably been left out, we believe that the ones included in the table are general enough to represent the characteristics of most of the cases. In the next section we will show what kind of network support we need so that such models can be effectively supported by computer-based multimedia tools.

3 Model of Interaction

We will now switch our attention from the physical world of interpersonal communication, to the virtual world of electronically supported, multimedia interaction. A number of MIM applications have been proposed to implement and support the communication models listed in Table 1. In addition to those, other applications have been presented, that do not have an exact correspondent in the traditional examples of social and professional interaction.

As we have seen in the previous section, 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 in the previous section.

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 MIM 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 the MIM 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* [8]. An example of this kind of application is *teleprogramming* [7], 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. For example, in a class lecture students are expected to ask for permission to interrupt the professor. In such cases, those rules and agreements dynamically modify the subset of participants authorized to transmit data. In the lesson example, the student who is authorized to intervene will become the speaker and the multimedia data produced by him will be transmitted to the other participants in addition to or in alternation with the data flowing from the professor.

An important example characterized by this model of interaction is the one generally classified as CSCW, or *computer supported collaborative work*. The goal of this service is to support the complex interactions required to allow effective meetings among geographically distributed professionals. In addition to video and audio data,

Type	Description	Model of Interaction	Data Flow	Accessibility	Event Scheduling
CSCW	All participants actively send/receive data	Dynamic	N -> N	Controlled	Planned
Telepresence	Many senders, one receiver	Static	N -> 1	Controlled	Unplanned
Virtual Cafe	All participants actively send/receive data	Dynamic	N -> N	Uncontrolled	Unplanned
Broadcasting Services	One sender many receivers	Static	1 -> N	Controlled	Planned
Distributed Computing	All participants actively send/receive data	Dynamic	N -> N	Controlled	Unplanned

Table 2: Characteristics of Selected MIM Applications.

participants in this model of interaction share applications such as editors, spreadsheets and drawing spaces. Information flows according to a highly dynamic, all-to-all pattern. Since the target environment is often that of technical and commercial enterprises, access control has usually to be enforced.

Another example is the so-called Virtual Café. The idea is to allow users at different sites to communicate informally through an electronic common space [2]. However, due to the informal nature of this model, access control is generally not of concern and scheduling is usually limited to the specification of some "opening hours".

Table 2 lists these new applications. As with Table 1, the entries are meant to indicate the dominant cases, but should not be taken as discriminative in the mathematical sense. Two more examples have been added to this table: *broadcasting services* and *distributed computing*. Even though they do not directly fall into the definition of MIM applications, we included them for their relevance in terms of network support requirements.

4 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

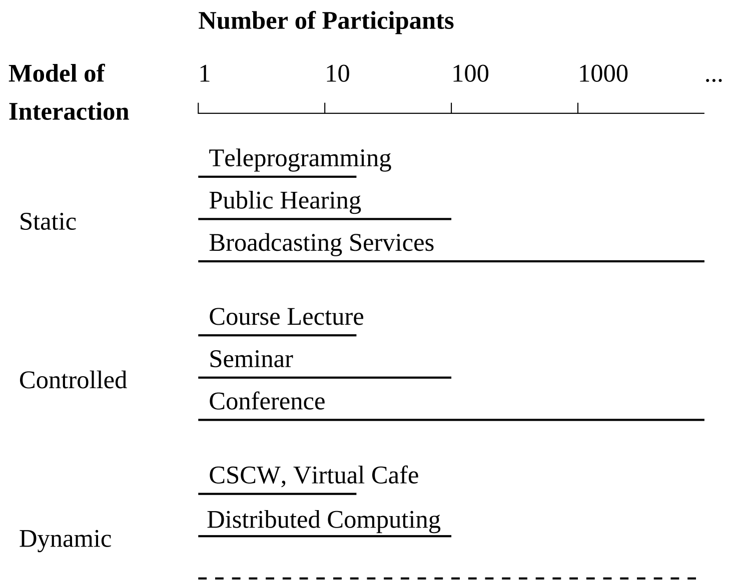

Figure 1: Size Ranges of MIM Applications.

be taken into account [12]. 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 the previous two sections (Tables 1 and 2). 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.

5 Requirements of MIM Application Clients

We can now try to list the MIM clients' wish-list, i.e. the list of services that a client would like to obtain from the network service provider, in order to access and manage MIM applications:

- *Joining and Leaving Applications*: Clients will dynamically join or leave MIM applications. Therefore, mechanisms should be provided by the network to allow such dynamic behavior of clients, for example to contact the application organizer beforehand to get an invitation if possible. For this reason, a *directory of services and applications* should also be available from the network.

- *Conference Organization and Management*: This service should permit a real-time network client to set up, schedule, reserve in advance, advertise and manage its own MIM applications. Management should include mechanisms for access control, authentication, invitation and initiation.

- *QoS Mechanisms*: MIM applications involve the coordination of a number of different media, each one of them characterized by its own QoS specifications. In some cases, a separate set of specifications might be allowed to each participant. The network should provide mechanisms for: *traffic requirement specification* (e.g. minimum and average throughput, burstiness), *performance requirement specification* (e.g. delay and jitter bounds), *synchronization control* (e.g. among destinations and among different media), *reliability specification* (e.g. loss probabilities and fault-tolerance requirements).

- *Scalability and Optimization*: MIM applications involve a number of different media, connecting a potentially large number of participants. Due to the large scale of some of the applications, some optimization in the allocation and management of resources is required, including the availability of load-balancing capabilities.

- *Security Mechanisms*: Mechanisms should be provided to help network clients manage security and accessibility of MIM applications.

All the above mentioned functionalities have to be considered as the basis for the design of a client-network interface; some of these functionalities, however, are intended to be accessible only to a subset of the application participants or to the network administrators. To get a better handle on this, a simple classification of the various foreseen interfaces is needed, where functions are considered as part of the same interface, if permissions to use one of the interface functions implies permission to use all of the interface functions. The terms *client* and *provider* are then defined relative to an interface. Thus, classifying interfaces and focussing on only part of these allows us to concentrate on a coherent subset of all the functions to be found in a monolithic client-network interface.

We consider three dimensions for interface classification purposes.

- *Layers of Abstraction*: Following the OSI Reference Model we assign each function to a certain layer. For this paper, layers of interest are *Network* and *Transport*.

- *Levels of Size*: According to the structure of large internetworks, we distinguish between the levels *Application-Specific*, *Network-Specific*, and *Internetwork-Specific*. It is important to understand that, say, "application-specific" does not mean that interfaces falling into this category are dependent on a particular application, but that each individual application is expected to make a specific use of such an interface by passing application-specific parameters. Likewise, a network-specific function is used with a single set of parameters network-wide. Such functions are meant to support management tasks such as system-tuning. Finally, internetwork-specific functions have common parameters for all networks connected by the same internetwork.

- *Separation of Functions*: In most of the proposed real-time protocols data handling and control entities are mixed in a single stack of abstractions. Since the control capabilities requested from a protocol providing guaranteed communication are increasing in complexity, we propose a clear separation of *Data* and *Control*, extending the model proposed in [1]. For example, aspects of data representation, fragmentation, and re-assembly are all data-related functions, while channel establishment, admission control, and dynamic joining/leaving are control-related functions.

In the subsequent sections we will concentrate on the interfaces that belong to the *control-related application-specific* subset. To arrive at a feasible and practical layer decomposition, we start by defining abstractions that directly relate to our domain analysis above. Thereafter, we consider some implementation alternatives and finally propose concrete interface parameterizations for the lowest three layers.

6 Multimedia Channels

In order to approach the design of concrete interfaces, we need to define abstractions well suited to the requirements of the applications we analyzed in the previous sections. All definitions introduce various forms of *channels*, where we consider a channel to be a logical link between information sources and information sinks over which communication can occur. The definitions use two techniques to derive new abstraction: The *addition of parameters* and the *composition of multiple (more primitive) channels* into a single, more powerful one. The former leads to definitions that are more concrete (and less general), while the latter defines channels that are more powerful (and less specific).

Definition 1 (Logical Channel) *A* logical channel *c in a network connecting a set of nodes $\mathcal{N}$, is a pair (σ, ρ), where $\sigma(c, n, t)$ and $\rho(c, n, t)$ are two predicates such that:*

- *a node $n_i \in \mathcal{N}$ is said to be a* channel sender *at time t iff $\sigma(c, n_i, t)$ holds;*
- *a node $n_i \in \mathcal{N}$ is said to be a* channel receiver *at time t iff $\rho(c, n_i, t)$ holds.*

The abstraction above defines a very simple form of communication involving a number of sources and of destinations of data. No specification is given regarding the quality of the communication service provided by means of such an abstraction. However, a real-time communication is actually characterized by a specification of the performance of the communication [4]. For this reason we now introduce the concept of real-time channel as a logical channel for which a characterization of its performance can be specified:

Definition 2 (Real-Time Channel) *A* real-time channel *r in a network connecting a set of nodes $\mathcal{N}$, is a refined* logical channel *(σ, ρ), where $\sigma(r, n, p, t)$ and $\rho(r, n, p, t)$ are two predicates such that:*

- *a node $n_i \in \mathcal{N}$ is said to be a* channel sender *at time t with performance p iff $\sigma(r, n_i, p, t)$ holds;*
- *a node $n_i \in \mathcal{N}$ is said to be a* channel receiver *at time t with performance p iff $\rho(r, n_i, p, t)$ holds.*

The real-time channel defined above is meant as a single media, virtual connection among a set of network nodes and it is intended to be an extension of the definition of real-time channel as originally introduced in [6]. Each node participates to the connection as a source or as a sink of data, and is characterized by its own performance specification. For example, a sender may be specified to have certain traffic characteristics, while a receiver may be specified to have certain QoS requirements.

These definitions try to individuate a first class of roles, i.e. *senders* and *receivers* of a real-time connection. The introduction of time in this definition is required to express the fact that, in general, a real-time channel is dynamic, in the sense that a participant can be source or destination at different time instants during the channel lifetime.

Since MIM applications usually involve a number of different media, we can now introduce an abstraction able to encapsulate this capability:

Definition 3 (Multimedia Channel) *A* multimedia channel m *is a triple ($\mathcal{M}$,ω,μ) where $\mathcal{M}$ is a finite set of real-time channels $\{r_1 \dots r_M\}$ and $\omega(m,n,t)$ and $\mu(m,n,t)$ are two predicates such that:*

- *a node $n_i \in \mathcal{N}$ is said to be a* multimedia channel organizer *at time t iff $\omega(m, n_i, t)$ holds;*
- *a node $n_i \in \mathcal{N}$ is said to be a* multimedia channel manager *at time t iff $\mu(m, n_i, t)$ holds.*

We can now try to use the formalisms presented above by applying them to the classification we introduced for MIM applications. If we consider the model of interaction, i.e. how differences in the applications influence the kind of communication service these applications require from the network, we can devise the following classification for a real-time channel r:

- *Static*: According to the earlier definitions, an application is characterized by a static model of interaction when both predicates $\sigma(r,n,p,t)$ and $\rho(r,n,p,t)$ are independent of time, i.e.

  $\sigma(r,n,p,t) \ = \ \sigma(r,n,p), \ \ \rho(r,n,p,t) \ = \ \rho(r,n,p).$

- *Controlled*: In this case the source and destination predicates are related: $\sigma(r,n,p,t) = f(\mathcal{R}, \rho(r,n,p,t))$ where $\mathcal{R}$ is determined by application dependent rules. For example, in the case of remote lecturing, we have supposed that there can be only one speaker at each time. Consequently, the relation can be simply expressed as:

  $n \neg n^\star \ \wedge \ \sigma(r,n^\star,p^\star,t) \ \Rightarrow \ \neg\sigma(r,n,p,t).$

- *Dynamic*: In this case the relation between $\sigma(r,n,p,t)$ and $\rho(r,n,p^\star,t)$ is very simple:

  $\sigma(r,n,p,t) \ = \ \rho(r,n,p^\star,t).$

7 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 [5].

We have individuated two main reasons for adopting such solution. The first one is that mechanisms for an efficient implementation of multicast communication begin to be available in a number of packet-switching 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 [6]), 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

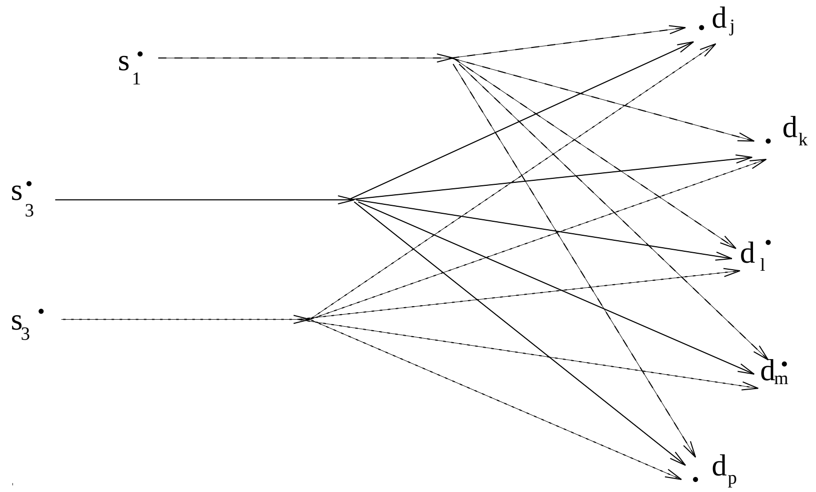

Figure 2: Possible implementation of a multimedia channel.

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

[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.

set is statically determined at the beginning of the application, we can adopt two different approaches.

8 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 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. [18]).

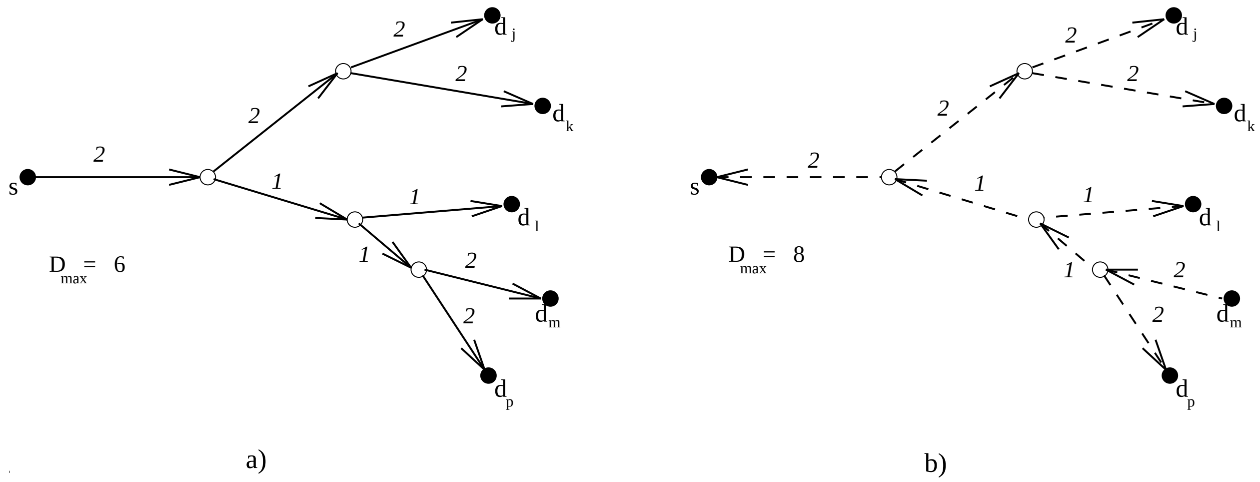

Figure 3: A half-duplex multicast connection.

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 fixed, 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.

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 (Cfr. 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

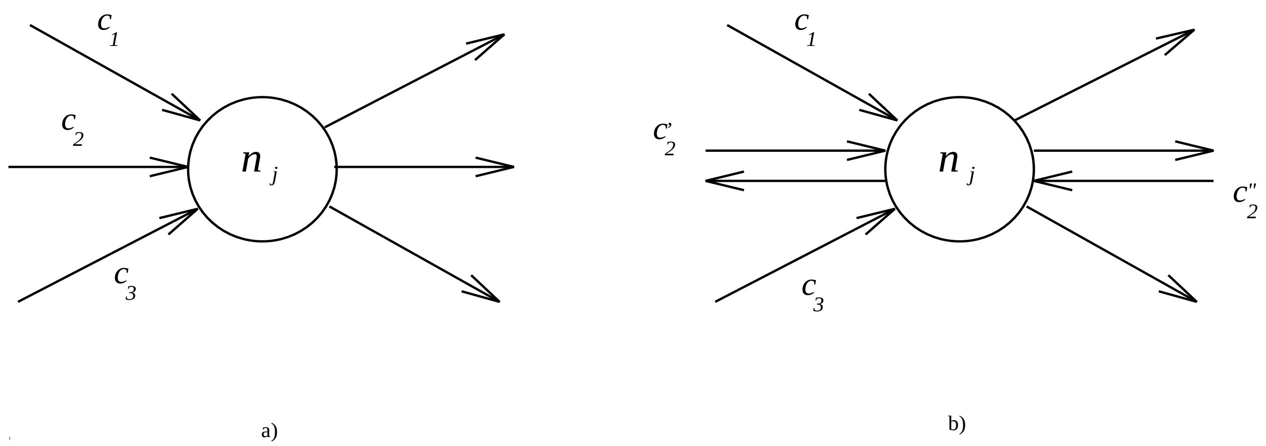

Figure 4: A possible solution for half-duplex connections.

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.

9 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 [6]. 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*.

9.1 The Deterministic Test

According to the Tenet scheme, the traffic requirements of a connections are expressed in terms of the minimum inter-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 $x_{min,2} = min(x_{min,2'}, x_{min,2''})$.

9.2 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 sufficient 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.

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

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.

9.3 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.

10 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 intend is to provide a building block that allows *session* or higher layers to compose needed abstractions like the *multicast channel* of section 6. 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. As a general principle, QoS parameters are specified by ranges where the desired value and the maximally tolerable deviation is 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 [6].

10.1 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.

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*}[4])
    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 = (*Opening-Bound*, *Closing-Bound*)

[3] marginal values are considered to be inclusive

$Opening\text{-}Bound = (T_{min}, \Delta T_{min})$
$\rightsquigarrow$ opening time (in seconds after some T_0) $T \in [T_{min}, T_{min} + \Delta T_{min}]$
$Closing\text{-}Bound = (t_{max}, \Delta t_{max})$
$\rightsquigarrow$ closing time (in seconds after some T_0) $t \in [t_{max} - \Delta t_{max}, t_{max}]$

$Indications = (Sharing\text{-}Indication,\ Priority\text{-}Indication)$
$Sharing\text{-}Indication = sharing\text{-}id$
$Priority\text{-}Indication = prio\text{-}level$

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*: The advance reservation interval is specified using four parameters (grouped into $Schedule\_Req$) that pin the opening time T and closing time t plus allowable deviations (ΔT and Δt). 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: $[T, t) = \bigcap_{i=1}^{N} [T_i, t_i) \subseteq [T_{min}, t_{max})$.

- *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.

10.2 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 transport layer creates a routing table and makes it available to the network layer. Also, the network layer implements the packet queueing and scheduling mechanism. For this purpose, the transport layer has to pass the appropriate information, e.g. the local processing deadlines and the maximum number of buffer-space to be used by the channel. Note that the network layer implements rate controlling by making sure that the channel does not exceed the assigned resources and by dropping packets otherwise. This is important to ensure proper protection of other real- and non-real-time channels from a malfunctioning real-time channel client [6].

11 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 existing support for simplex multicast communication characterizing recent proposals for packet-switching 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 devising mechanisms such that, within certain bounds, the specification of quality of service requirements can be allowed to all the nodes interconnected by a half-duplex channel.

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.

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.

Appendix A

The terminology we propose here has been derived from the one usually adopted for the definition and characterization of typical patterns in the area of interpersonal communication [15].

- *Conference*: Even though the term *conference* is generally associated to a specific type of multiparty interaction, we will interchangeably use both terms *MIM application* and *conference* to identify applications involving a number of participants exchanging multimedia data under real-time constraints.

- *Participant*: The parties involved in a conference are called *participants*. Each participant is an addressable network entity related to the conference in at least one point in time. A participant typically is a human being; however, we believe that our classification might be extended to consider also non-human participants, as is the case of some multimedia applications such as teleprogramming [7] and multimedia database access [14]. A basic criterion could be that the participants involved are assumed to be subject to real-time constraints, and are, in some sense, "interactive". Involvement in a conference is understood here in a rather general sense; e.g., it includes related activities taking place either before the conference's starting time or after its conclusion.

- *Role*: The participants relate to the conference by taking certain roles. Each role represents a contribution to the conference itself, its organization, or its management. In our opinion, four basic roles can be distinguished:

  Organizer (or Initiator): A participant who initiates/establishes the conference, decides on invitations, reserves in advance the resources needed to run the conference, specifies security and accessibility requirements, specifies the quality of service requirements (harmonizing them to the characteristics of the expected participants), and is responsible for paying the bill.

  Manager (or Chair): A participant who *runs* the conference, decides admission of late comers, moderates the discussion, makes decisions about the possible restructuring of the conference after admissions and departures, and closes the conference.

  Receiver: A participant who can access the multimedia data produced within the conference.

  Sender: A participant who can produce part of the multimedia data that may be consumed by the receiving participants of the conference.

Note that a role can be shared by multiple participants. For example, in some case all participants (or a subset) can take the manager's role, effectively making the conference management fully (or partially) distributed.

- *Role Association*: A participant may qualify for multiple roles. In principle, participants may request or give up a role at any point in time. A participant is defined to have a certain role, even if it is not actually playing that particular role all the time. The classification we propose is related to whether a participant is to be allowed to assume a role during its lifetime. If it is, proper communication support must be provided by the network. Of course, certain restrictions might apply, depending on the application's characteristics. For example, the number of participants assuming a certain role and the dynamics of role association may be limited by rules or agreements, explicitly or implicitly accepted by the participants themselves.

- *Media*: Conferences are based on participants communicating through the use of a number of media. To clarify the scope of this document, the following are the media being considered here:

  Motion Video

  Audio

  Still Images

  Image Streams

  Traditional Data, possibly to be distinguished further into:

    Low-Bandwidth, Interactive

    High-Bandwidth, Bulk-Transfer

References

[1] A. Banerjea and B. A. Mah. The Design of a Real-Time Channel Administration Protocol. In *Proceedings of Second International Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, November 1991.

[2] S.A. Bly, S.R. Harrison, and S. Irwin. Media Spaces: Bringing People Together in a Video, Audio and Computing Environment. *Communications of ACM*, 36(1), January 1993.

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

[4] D. Ferrari. Client Requirements for Real-Time Communication Services. *IEEE Communication Magazine*, 11(11), November 1988.

[5] 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.

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

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] M. Stonebraker and J. Dozier. Large Capacity Object Servers to Support Global Change Research. Technical Report 91/1, Sequoia 2000, Berkeley, 1991.

[15] C. Szyperski and G. Ventre. A Characterization of Multi-Party Interactive Multimedia Applications. Technical Report TR-93-006, International Computer Science Institute, January 1993.

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

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

[18] 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.