A Call Model For Multipoint Communication in Switched Networks

Michael E. Gaddis Rick Bubenik John D. DeHart

Applied Research Laboratory
Department of Computer Science
Washington University
St. Louis, Missouri 63130-4899

ABSTRACT We describe a *call model* for multipoint communication in switched networks. The model provides network clients with dynamic multipoint, multiconnection communication channels, which we term *calls*. Clients create, manage and manipulate calls using our Connection Management Access Protocol (CMAP). The call model provides basic interconnection services suitable for local and wide area networks, where more sophisticated services can be layered over this substrate.

1 Introduction

Multimedia workstations, characterized by video, audio and data transfer capabilities, are becoming increasingly popular. When several multimedia workstations are interconnected over a network, the bandwidth requirement is much higher than that with traditional data-only workstations. Shared media networks, such as Ethernet and FDDI, do not have the required bandwidth to support many of these multimedia applications, especially in cases where several multimedia workstations are connected to the same physical network, and where several workstations are communicating with one another simultaneously.

Switched Local Area Networks (LANs) offer the potential to alleviate this network bottleneck. With switched LANs, such as those based on the Asynchronous Transfer Mode (ATM) technology [4] [6] [15], each workstation has a dedicated link (in the case of ATM, at 155 megabits or higher). The workstations are interconnected through one or more *packet switches*, which allow simultaneous transfers at maximum link rate between several workstations. In addition to solving the immediate problem of supporting multimedia traffic in local areas, switched networks have the potential to scale to Metropolitan Area Network (MAN) and Wide Area Network (WAN) sizes. The ATM standard is being developed currently by CCITT for the public telecommunication WANs. As such, switched ATM LANs will likely be able to interface directly to the public networks, perhaps creating one ubiquitous global network.

This paper describes a *call model* designed for switched ATM networks applicable to both the LAN and WAN environments. The model supports multipoint, multiconnection communication channels, which we term *calls*. Calls are allowed to change dynamically during their lifetime, in terms of the number of participants, the number of connections and the bandwidth of the connections. Our model defines how clients view and interact with the network to create, manage and manipulate calls. It defines the outward functionality of the network, its expected behavior and atomic capabilities. Network clients interact with the network using our Connection Management Access Protocol (CMAP) [1] [11] [12].

The remainder of this paper is organized as follows. In Section 2, we give a brief overview of ATM networks. In Section 3, we describe the ATM standard and the basic data information exchange component, the ATM *cell pipe*. In Section 4, we describe two types of ATM connections, *virtual path* and *virtual channel*, and discuss how they can be used for multipoint communication. In Section 5, we describe the call model. Finally, we cover related work in Section 6 and summarize in Section 7.

2 Overview of a Switched ATM Network

An example ATM network is shown in Figure 1. The network consists of *clients*, *exterior nodes* (nodes that interface to clients), and *interior nodes* (nodes that interface to other nodes only), all interconnected by *fiber optic links* operating at 155 Mbps or higher. A client signals the network to set up connections with other clients by sending *control* messages to exterior nodes, which carry out the requested client operations. If the request requires the services of other nodes, then control messages are exchanged between nodes (exterior and interior) in order to satisfy the request.

Each node in the network contains one or more ATM switches. The switches route each ATM cell (packet) to the desired destination link(s) based upon header fields in the cell (Section 3). In order to keep up with line speeds, the switches perform all routing in hardware. Since the time interval within which each cell must be routed is very small, tables in the switches are preconfigured with routing information. This makes ATM networks more suitable for connection oriented traffic, where the switch tables can be configured during connection setup. Connectionless traffic can be accommodated via *overlay* networks utilizing special purpose routers or datagram processors [2] [5].

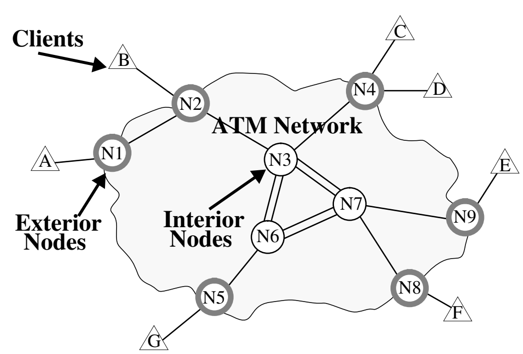

Figure 1. Example ATM network.

From: *ICC '92: Proceedings of the IEEE International Conference on Communications*, pages 609-615, June 1992.

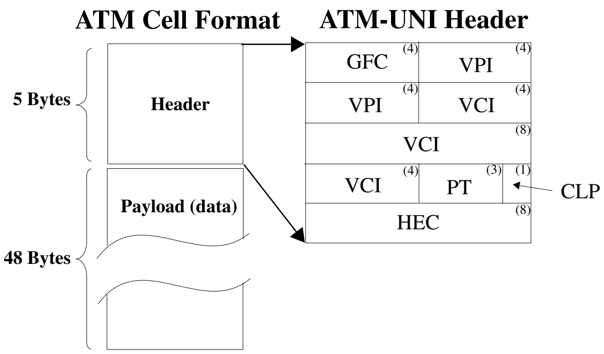

Figure 2. ATM UNI Cell Format.

3 The ATM Standard and Cell Pipes

The emerging ATM standard [4] [6] specifies link level cell formats for two interfaces: 1) the User-Network Interface (UNI), for communication between the client and the network, and 2) the Network-Network Interface (NNI), for communication between network nodes. The ATM UNI cell format is shown in Figure 2. It consists of a 48 byte information (data) field and a five byte header. The header has six fields: a Global Flow Control (GFC, 4 bits), a Virtual Path Identifier (VPI, 8 bits), a Virtual Channel Identifier (VCI, 16 bits), a Payload Type (PT, 3 bits), a Cell Loss Priority (CLP, 1 bit) and a Header Error Check (HEC, 8 bits).

Clients of ATM networks communicate over what we term *cell pipes*. Our model supports n-way, bidirectional, multipoint cell-pipes, where point-to-point channels are simply a special case of multipoint channels. Figure 3 shows the conceptual view of a three endpoint cell pipe between clients A, B and C. All data sent by one client is received by all other clients who have elected to receive on this cell pipe.

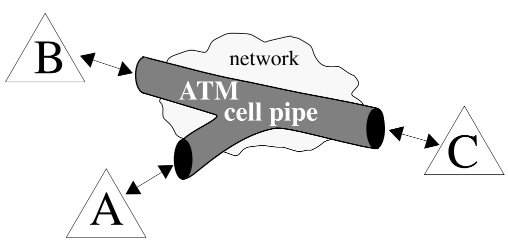

Figure 3. Three-way Cell Pipe.

Cell pipes are implemented in the ATM network by setting up low level connections between each switch traversed by the cell pipe. When a cell arrives at a switch, its VPI/VCI is examined and used to route the cell to the appropriate output link. The VPI/VCI is then updated so as to be meaningful to the next switch on its route. When more than two clients communicate over a cell pipe, cells have to be duplicated at one or more switches so that a copy of each cell arrives at all endpoints.

4 Virtual Path and Virtual Channel Connections

In our call model, we allow clients to designate whether they want a *virtual path* (VP) or a *virtual channel* (VC) connection at connection setup time [3]. With VP connections, the network uses only the VP for routing at each intermediate node and, thus, does not translate the VCI (delivering it unchanged from the value assigned by the sender). This allows clients to use the VCI for source discrimination in multipoint connections, provided that each client places a unique VCI in all of its outgoing cells. (We leave the coordination of VCI assignment to the clients.)

In Figure 4, we expand the view presented in Figure 3 of the three endpoint multicast call to highlight the most general case allowed. The example shows three endpoints, A, B and C, communicating using a VP connection. Each endpoint is receiving and transmitting on the connection and all cells sent by a transmitter will arrive in the same order at each receiver. In the example, the clients have agreed that VCI 1 would be used for client A, 2 for B and 3 for C. Therefore, by using the VCI, all clients have the capability to distinguish the source of the transmitter.

With VC connections, the network uses the VCI to route the cells and may also translate the VPI on internal links (for inter-trunk grouping, where multiple VC connections are carried on a preconfigured internal trunk). The VCI used by a transmitter, therefore, has no end-to-end significance. At some point during communication, all transmitters on a VC connection may have their cells interleaved. The cells arriving at a receiver will not be distinguishable by the ATM cell header—the VPI and VCI are the same for all arriving cells regardless of origin. Therefore, if

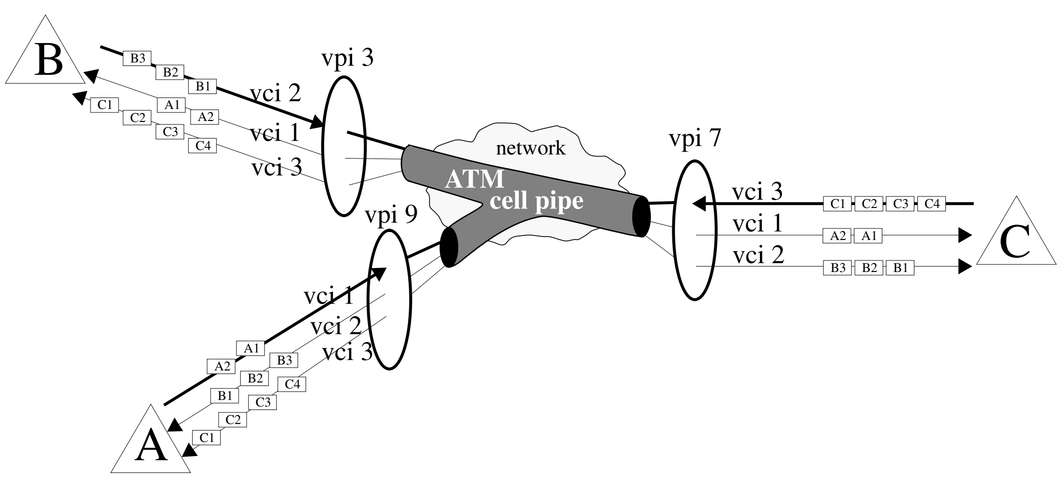

Figure 4. Sequenced Bidirectional Multipoint Virtual Path Connection Using the VCI for Source Discrimination.

VC connections are used for multipoint communication, higher level protocol information embedded in the ATM cell payload must be used for source discrimination (if required). VC connections are desirable for connections that do not need source discrimination, and for connections that want to take advantage of rapid setup (where the network is able to reduce connection setup overhead by using preconfigured trunks).

5 Call Model

Our approach to the development of a call model for switched ATM networks is derived from a layered networking approach, as shown in Figure 5. We provide mechanisms for the use and maintenance of ATM network connections, but do not directly integrate higher level services into the model (such as NTSC to HDTV video conversion). Our model views all clients as *equal*. We draw no distinction between network gateways, multimedia workstations or video phones. All clients are constrained to access the network through our CMAP protocol and communicate through ATM cell pipes. We believe that our call model provides the necessary core functionality so that other services can be layered over CMAP, perhaps presenting a more sophisticated network model to users.

Network clients communicate with other clients by creating and manipulating *calls*. A call is a distributed object maintained by the network that describes the communication paths that interconnect clients. Initially, at call creation, a call has one or more communication channels, which we term *connections* (a connection is an end-to-end cell pipe). The owner of a call (the creator) can add connections to or remove connections from the call. Multiple connections within a call are useful for applications such as video conferences, where one connection carries video and another audio. When a call is first created, the owner (and possibly one additional endpoint) are the only endpoints in the call. Additional endpoints may be added by: 1) invitation from the owner, where the invited party has the option of refusing the invitation, 2) request from a client not currently in the call to be added, where the owner has the option of denying the request, or 3) request from a third party, not necessarily in the call, to add a client, where both the owner and the client being added have the option to refuse. Our call model is dynamic, in that the number of endpoints and number of connections in the call can vary over time, while communication is taking place. Additionally, the bandwidth reserved by a connection can also increase and decrease.

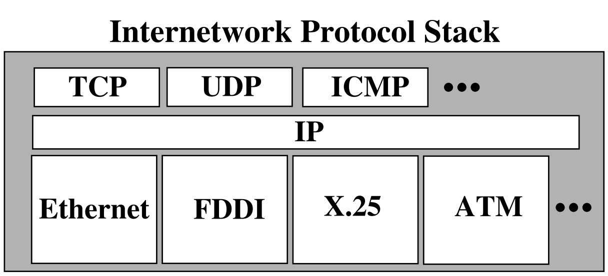

Figure 5. ATM as a Component Network in an Internetworking Environment.

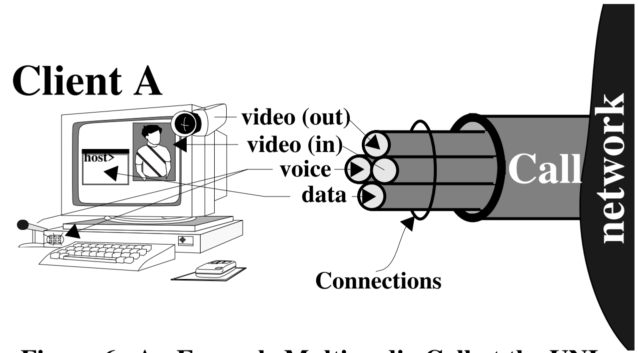

Figure 6. An Example Multimedia Call at the UNI.

5.1 Calls

The call is the primary object manipulated by clients. When a client adds himself to a call he gains access to all connections of the call. Figure 6 shows an example of a multimedia call at a client. Calls have a number of parameters that describe how clients may access the call, described in the following subsections.

5.1.1 Call Type

The call type designates whether the call is *multipoint* or restricted to *point-to-point*:

$$\textbf{Type} \in \{\textbf{MULTIPOINT}, \textbf{POINT-TO-POINT}\}$$

While the call model supports multicast calls as the general object, some calls are intrinsically point-to-point and, if designated as such, can be serviced more efficiently by the network (for example, by routing the call over internal trunks).

5.1.2 Call Accessibility

The accessibility parameter is a 1-tuple that describes how clients are allowed to join the call:

$$\textbf{Accessibility} \in \{\textbf{OPEN}, \textbf{VERIFY}, \textbf{CLOSED}\}$$

If the call is OPEN, any client in the network may join the call without the owner's permission. A CLOSED call restricts the call such that only the owner may add clients. If the call has the VERIFY accessibility, then any client may attempt to join the call, but the operation will be verified with the owner before it is allowed to complete.

5.1.3 Call Monitoring

The monitoring parameter is a 3-tuple designating whether endpoints of the call are notified when new endpoints join or drop out of the call:

$$\begin{aligned}\textbf{Monitor} = \langle\, &\textbf{owner} \in \{\textbf{ON}, \textbf{OFF}\},\\ &\textbf{transmitters} \in \{\textbf{ON}, \textbf{OFF}\},\\ &\textbf{all} \in \{\textbf{ON}, \textbf{OFF}\} \,\rangle\end{aligned}$$

If the *owner* field is set to ON, all endpoint joins and drops are reported to the owner. Likewise, if the *transmitters* field is set to ON, all transmitters are notified, and if the *all* field is set to ON, all participants in the call are notified. When used in conjunction with an accessibility of OPEN, an owner modifiability setting of ON allows the owner to keep track of endpoint joins and drops without being burdened with explicitly verifying every such change.

5.1.4 Call Modifiability

The modifiability parameter controls whether non-owners have the permission to add connections to the call:

$$\mathbf{Modifiability} \in \{\mathbf{ON}, \mathbf{OFF}\}$$

One example of where this option is useful is in multiparty conference calls, where each endpoint adds a one-to-all connection for his feed when joining the call.

5.1.5 Call Priority

The priority parameter is a 1-tuple that allows some calls to take precedence over others:

$$\mathbf{Priority} \in \{\mathbf{NORMAL}, \mathbf{PREEMPT}, \mathbf{OVERRIDE}\}$$

NORMAL is the lowest priority, followed by PREEMPT, followed by OVERRIDE. If the network does not have the resources to support a higher priority call and the required resources are currently allocated to lower priority calls, these resources are reclaimed (aborting the lower priority calls) so that the higher priority call can be supported. The intention is that most clients would only be allowed to specify normal priority calls, while selected clients (for example, the national weather service or the military) would be given the option of using one or both of the other priorities (as configured by network management).

5.1.6 Call Connection List

The connection list describes the connections of the call. The description includes allocated bandwidth, receive/transmit permissions, and connection type, as described in the following section. The list is dynamic since connections can be added to or removed from the call at any time, and since the attributes of connections can change over time.

5.2 Connections

The connection is the primary information carrying component of a call. The five parameters associated with all connections are described in the following subsections.

5.2.1 Connection Type

The connection type is a 3-tuple:

$$\begin{aligned}\mathbf{ConnType} = \langle\ & \mathbf{channel\_type} \in \{\mathbf{VP}, \mathbf{VC}\}, \\ & \mathbf{BW\_type} \in \{\mathbf{STATIC}, \mathbf{DYNAMIC}\}, \\ & \mathbf{QOS} \in \{\mathbf{HIGH}, \mathbf{MEDIUM} \text{ or } \mathbf{LOW}\}\ \rangle\end{aligned}$$

The first option specifies the type of channel that the connection requires and must be one of two values: *virtual path* (VP) or *virtual channel* (VC), see Section 4. The second option specifies whether the connection is DYNAMIC or STATIC. If a connection is *static* then the bandwidth is fixed throughout the life of the call, which makes the connection more predictable and allows more efficient use of network resources. DYNAMIC connections can have their bandwidth modified during the life of the call. The last type option specifies the *quality-of-service* (QOS) desired and may have the values of HIGH, MEDIUM or LOW. QOS relates to options for cell loss behavior that may vary from network to network. QOS, therefore, is left intentionally vague. If the network can implement different cell loss behavior strategies then the network control software will group these into the categories of HIGH, MEDIUM and LOW.

5.2.2 Connection Permissions

Connection permissions describe the allowed receive and transmit capability of endpoints on the connection. Each connection has a *default* permission, which can be overridden by the call owner so that individual endpoints can be assigned different permissions. The permission tuple consists of a receive part and a transmit part, as shown below:

$$\begin{aligned}\mathbf{Perm} = \langle \mathbf{Rcv} &= \langle\ \mathbf{perm} \in \{\mathbf{ON}, \mathbf{HOLD}, \mathbf{OFF}\}, \\ &\qquad \mathbf{changeability} \in \{\mathbf{ON}, \mathbf{VERIFY}, \mathbf{OFF}\}\rangle, \\ \langle \mathbf{Trans} &= \langle\ \mathbf{perm} \in \{\mathbf{ON}, \mathbf{HOLD}, \mathbf{OFF}\} \\ &\qquad \mathbf{changeability} \in \{\mathbf{ON}, \mathbf{VERIFY}, \mathbf{OFF}\}\rangle\rangle\end{aligned}$$

For the receive part, a *perm* of ON indicates that the client can receive on this connection, a *perm* of OFF indicates that the client cannot receive on this connection, and a *perm* of HOLD indicates that the client cannot receive, but that bandwidth should be reserved within the network up to the client's access node (for use when the client's receive permission is to be changed to ON at some later time). The *changeability* field indicates whether a client can deviate from his assigned permissions when mapping the connection in, as described in the next section. The transmit part is defined analogously to the receive part.

5.2.3 Connection Mappings

The connection mapping is set by the client and determines how the client accesses the connection. The mapping tuple is defined as follows:

$$\begin{aligned}\mathbf{Mapping} = \langle\ & \mathbf{Receive} = \langle\ \mathbf{perm} \in \{\mathbf{ON}, \mathbf{HOLD}, \mathbf{OFF}\}\rangle, \\ \langle\ & \mathbf{Transmit} = \langle\ \mathbf{perm} \in \{\mathbf{ON}, \mathbf{HOLD}, \mathbf{OFF}\}\rangle, \\ \langle\ & \mathbf{Echo} = \langle\ \mathbf{perm} \in \{\mathbf{ON}, \mathbf{OFF}\}\rangle\rangle\end{aligned}$$

The connection mapping is constrained by the connection permission. If the permission *changeability* field is set to OFF for the receive (transmit) part, the client's receive (transmit) mapping must equal his permission *perm*. If *changeability* is ON, then the client may choose any mapping. In this case, the permission is not enforced and is merely viewed as a suggestion to the client. If the *changeability* is VERIFY, the client can alter his mapping, but the owner will first be queried to see if the new mapping is acceptable. There is one exception to this general rule: if a receive or transmit permission *perm* is ON, the client can freely change to HOLD regardless of the *changeability* setting.

The *echo* mapping is used in conjunction with transmit to allow clients to view their own transmissions. When set to ON, the client's transmissions will be echoed and when set to OFF they will not be echoed. Clients who are allowed to transmit can freely change their echo mapping.

5.2.4 Connection Bandwidth Specification

We support two types of bandwidth specification for connections: 1) *reserved*, where the network guarantees and enforces

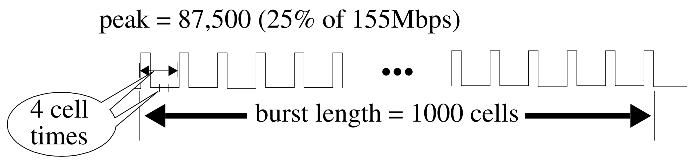

Figure 7. Peak and Burst Length at the UNI.

the requested bandwidth, and 2) *best-effort*, where the network neither guarantees nor enforces. Reserved bandwidth connections are treated preferentially by the network, whereas best-effort connections compete with one another for the remaining bandwidth not currently used by reserved connections.

The connection's bandwidth is specified as a 3-tuple, as shown below:

Bandwidth = < peak, average, peak_burst_length >

The *peak* and *average* parameters are expressed in cells per second. The *peak burst length* is measured in cells and indicates the maximum number of cells that the client can send during a burst at peak rate. Best-effort connections are indicated by all zeros in the bandwidth specification.

For reserved bandwidth connections, the figures given are for aggregate bandwidth, as seen at any receiver in the connection (this takes into account the increased loading that results when the transmissions from multiple transmitters combine and flow over links traversed by the connection). Figure 7 illustrates how we use the bandwidth specification to characterize the client's transmit behavior on a connection at the UNI. The *peak* value tells us a percentage of the bandwidth used by the client when transmitting a burst of data. The user may not arbitrarily send cells back-to-back for the duration of the *peak burst length*, but must *pace* them according to the *peak* value as shown. The *peak burst length* gives the network an indication of how long the connection will remain transmitting at *peak* rates. The example in Figure 7 shows a *peak* value of 87,000 cells per second (in this case, roughly 25% of a 155Mbps link) and a *peak burst length* of 1000 cells. The *average* value is used over a longer period of time to enforce the long term average utilization of the connection.

For best-effort connections, the client is not restricted in any way. Instead, all cells are forcibly marked at the UNI by setting the CLP bit. Inside the network, as buffers fill, the network discards marked cells if there is no room in the buffers for unmarked (reserved) cells.

We chose to provide both reserved and best-effort bandwidth connections for several reasons. Some applications clearly need bandwidth guarantees, such as live video, where cell loss can result in incomprehensible transmissions. However, many other applications do not need these guarantees. For example, data transfer applications can usually tolerate lossy networks and network congestion using schemes such as retransmission and backoff, as in TCP/IP [10]. Additionally, many applications cannot easily predict their bandwidth requirements, and if forced to do so would produce specifications that are too low or too high, each with negative consequences. When all applications are forced to use reserved, guaranteed connections, network utilization is sometimes low (depending on the connection mix) since the level of allowed multiplexing must be limited to provide the guarantees. By introducing the best-effort bandwidth class, we accommodate those connections that do not need the guarantees and cannot easily estimate requirements, and we provide a means whereby the excess capacity of the network can be utilized.

5.2.5 Connection Notification

The notification parameter is set by each client when joining a connection and has one of two values:

$$\textbf{Notification} \in \{\textbf{ON}, \textbf{OFF}\}$$

When notification is *on*, the client will be notified of all changes in bandwidth that occur on the connection. If *off*, the client is not notified. This parameter is useful when a client wishes to reconfigure an interface (for example, by adding more buffers) when a bandwidth change occurs.

5.3 VPI/VCI Assignment at the UNI

Our call model supports both *uni-directional* and *bi-directional* assignment of VPIs and VCIs at the UNI. Uni-directional assignment allows the VPI and VCI to differ for each direction of a *receive/transmit* connection. Bi-directional assignment requires the VPI/VCI values to be the same for both directions. In the case of uni-directional assignment, the receiving party on a link is responsible for assigning the VPI/VCI values that the sending party is to use. This policy was chosen since the tables that decode VPIs and VCIs in arriving cells are typically located at the receiver's site and managed by the receiver. When setting up a connection, a network entity (client or node) allocates a VPI/VCI table entry and informs the adjacent entity that he is to use the assigned VPI/VCI when sending cells to it. This resource management information is sent with the call/connection request. The adjacent network entity indicates the VPI/VCI for the opposite direction when it acknowledges the operation. Each entity allocates the resources according to this algorithm, obviating the need for VPI/VCI assignment collision detection, avoidance and recovery.

At the UNI, we allow clients to specify the VPI/VCI pair for both directions. In this way, a bi-directional assignment can be made by choosing identical values for each direction. The client can also leave both pairs empty, in which case the network will assign both (choosing identical values, if a strictly bi-directional assignment policy is in force).

5.4 Signaling the Network

Clients create, manipulate and delete calls by signaling the network using our Connection Management Access Protocol (CMAP) [1] [11] [12]. CMAP messages are sent in-band over the ATM link connecting the client to the network. CMAP is layered over a transport service, referred to as CTL (CMAP Transport Layer), that provides reliable delivery of CMAP messages, and segmentation and reassembly (SAR) of CMAP messages into ATM cells (Figure 8). In the CMAP specification [11], we define the general requirements for CTL,

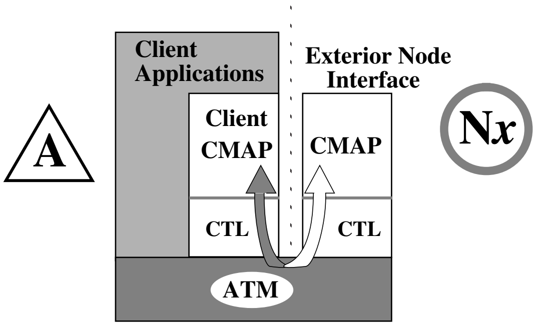

Figure 8. Client Access Protocol Stack for CMAP.

but do not stipulate a specific transport protocol or SAR protocol. As a result, CMAP can be layered over many existing protocols.

At the lowest level, the ATM cells comprising CMAP messages are sent over a well-known signaling connection, identified by a well-known VPI/VCI pair. One such connection is setup from each client to the switch node's control processor when the node bootstraps. Clients can request the creation of additional signaling connections, but are not required to do so.

While we expect that there will typically be one client connected to each ATM switch access link in the ATM LAN environment, CMAP allows for multiple clients on the same link, as shown in Figure 9. This is one case where additional signaling connections may be desirable, so as to differentiate the multiple clients on the access link.

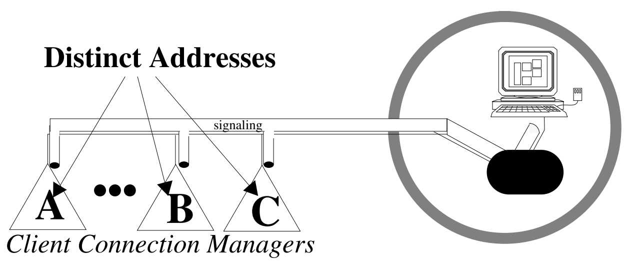

Figure 9. Multidrop Access at the UNI.

Another signaling capability that we allow for is *surrogate signaling*, where one client (called the *surrogate* client) is designated as the signaling entity for another client (called the *mute* client), as shown in Figure 10. The surrogate client originates all signaling messages for the mute client and receives all signaling messages from the network that would otherwise be sent to the mute client. Surrogate client signaling requires external configuration by network management. The *surrogate* client can be anywhere in the network and does not have to be local to the *mute* client's node.

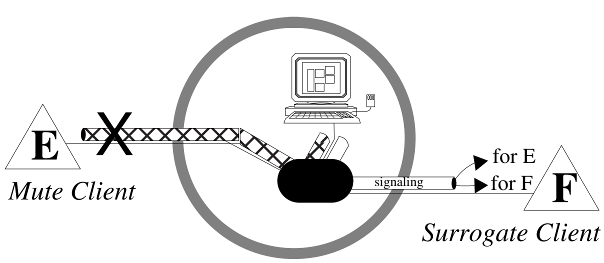

Figure 10. Surrogate Client Signaling at the UNI.

6 Related Work

Minzer and Spears [16] [17] discuss multipoint connections in *Broadband Integrated Services Digital Networks* (BISDNs). One difference between their view of a network and our view is that they believe higher level information processing functions, such as video format conversion, should be performed and managed by the network. We feel that the network should provide a flexible interface so that these functions can be easily provided as client-based services (where certain clients offer higher level services to other clients). Our approach simplifies network hardware and software functionality. Both models have good qualities and address issues from different perspectives.

Bellcore's Information Networking Architecture (INA) discusses many of the lower level connection management capabilities that are needed by the call management layer [13]. INA seeks to provide a substrate upon which many different call models and call control protocols can be layered. In this respect, our call model and CMAP become candidates for the INA call management layer, along with many others.

CCITT has begun to explore the requirements of call control and management in telecommunications networks for BISDN [6] [14]. Currently, it appears as if the CCITT standards will be extensions of existing protocols (such as Q.931), rather than new protocols designed explicitly for high speed ATM networks. As such, the CCITT protocols will likely offer the advantage of backward compatibility, but at the cost of complexity and a general mismatch of protocol/network capabilities.

7 Summary

We believe that switched network architectures based on fast packet switching technology are needed to support the growing demand for the reliable, high performance communication required by multimedia applications. We have presented a call model for switched LANs, MANs and WANs based on the ATM standard. The key features of our model are support for multipoint, multiconnection calls as a fundamental object and support for dynamic calls whose attributes (such as number of connections, number of endpoints, and connection bandwidths) can change over the life of the call. Our model supports: 1) point-to-point communications involving two clients, such as that used in simple telephone calls and data exchange, 2) many-to-many communications involving from a few to hundreds of clients, such as that used in teleconferencing and in multicast group communication [9], 3) one-to-many communications involving, perhaps, tens of thousands of clients, such as that used in broadcast video distribution, and 4) many-to-one communications, involving an arbitrary number of clients, such as that used in error logging and surveillance systems.

The call model was designed as part of the Fast Packet Project [7], which culminated in a four switch metropolitan area test bed with interconnection links operating at 100 megabit per second [18] [19]. We are continuing to refine the call model and implement CMAP as part of Project Zeus [8]. The goal of Zeus is to deploy a campus area high speed network consisting of commercial 155+ megabit per second switches for support of existing applications and new high speed application research.

References

[1] R.G. Bubenik, J.D. DeHart and M.E. Gaddis. “Multipoint Connection Management in High Speed Networks.” In *IEEE Infocom ‘91: Proceedings of the Tenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 59-68, April 1991.

[2] R.G. Bubenik, M.E. Gaddis and J.D. DeHart. “A Strategy for Layering IP over ATM”. Washington University Applied Research Laboratory, Working Note 91-01, Version 1.1, April 1991.

[3] R.G. Bubenik, M.E. Gaddis and J.D. DeHart. “Communicating With Virtual Paths and Virtual Channels.” In *IEEE Infocom ‘92: Proceedings of the Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies*, May 1992.

[4] ANSI T1S1 Technical Sub-Committee. *Broadband Aspects of ISDN Baseline Document*. T1S1.5/90-001, June 1990.

[5] Bell Communications Research. *Generic System Requirements in Support of Switched Multi-Megabit Data Service*. Technical Advisory TA-TSY-000772, Issue 3, October 1989.

[6] CCITT. *Report of Working Party XVIII/8 (General B-ISDN Aspects),* Study Group XVIII—Report R 70, July 1991 (Report on Geneva Meeting, 11–28 June 1991).

[7] J.R. Cox. “Overview of the Washington University Fast Packet Project”. Washington University, Applied Research Laboratory, Working Note 89-02, September 1989.

[8] J.R. Cox and J.S. Turner. “Project Zeus Design and Application of Fast Packet Campus Networks”. Washington University, Department of Computer Science, Technical Report 91-45, July 1991.

[9] D.R. Cheriton and W. Zwaenepoel. “Distributed Process Groups in the V Kernel.” In *Transactions on Computer Systems*, 3(2):77-107, May 1985.

[10] D. Comer. *Internetworking With TCP/IP Principles, Protocols, and Architecture*. Second Edition, Prentice Hall, 1991.

[11] J.D. DeHart, M.E. Gaddis and R.G. Bubenik, “Connection Management Access Protocol (CMAP) Specification,” Washington University, Department of Computer Science, Technical Report 92-01, January 1992.

[12] M.E. Gaddis, R.G. Bubenik, and J.D. DeHart. “Connection Management for a Prototype Fast Packet ATM B-ISDN Network.” In *Proceedings of the National Communications Forum*, vol. 44, pp. 601-608, October 8-10, 1990.

[13] L.J. Iacona, W.C. Phillips and D.R. Pomplun. “INA Cycle 1 Network Management Functional Architecture (Issue 1).” Bellcore Technical Memorandum TM-NWT-020239, December 20, 1991.

[14] M. Kawarasaki and B. Jabbari. “B-ISDN Architecture and Protocol.” In *IEEE Journal on Selected Areas in Communications*, 9(9):1405-1415, December 1991.

[15] S.E. Minzer. “Broadband ISDN and Asynchronous Transfer Mode (ATM).” In *IEEE Communications Magazine*, 27(9):17-24, September 1989.

[16] S.E. Minzer. “A Signaling Protocol for Complex Multimedia Services.” In *IEEE Journal on Selected Areas in Communications*, 9(9):1383-1394, December 1991.

[17] S.E. Minzer and D.R. Spears. “New Directions in Signaling for Broadband ISDN.” In *IEEE Communications Magazine*, 27(2):6-14, February 1989.

[18] J.S. Turner, “New Directions in Communications.” In *IEEE Communications Magazine*, 24(10):8-15, October 1986.

[19] J.S. Turner, “Design of a Broadcast Packet Switching Network.” In *IEEE Transactions on Communications*, 36(6):734-743, June 1988.