IP on ATM Local Area Networks

ATM offers increased bandwidth and greater flexibility and manageability. However, ATM's success as a LAN technology depends on its ability to provide LAN-like services compatible with existing protocols and applications.

H. Jonathan Chao, Dipak Ghosal, Debanjan Saha, and Satish K. Tripathi

One of the key developments in desktop computing during the last decade has been the explosive growth of computer networks. These networks offer a wide variety of services from simple e-mail to more complex services such as file transfers and remote login to supercomputer sites. Computer networks consist of local area networks (LANs) interconnected by gateways and routers to form wide area networks. The existing local area networks are primarily based on shared media interconnects, which are likely to become potential bottlenecks not only because of new multimedia applications but also because of the rapid growth of services employing simple data transfers. Asynchronous transfer mode (ATM), a switching and multiplexing standard for broadband integrated networks, is viewed as an emerging technology capable of removing this bottleneck. Much of the interest in ATM stems from the promise of its vastly increased bandwidth and greater flexibility and manageability. However, its success as a LAN technology depends on its ability to provide LAN-like services compatible with existing protocols and applications. In this paper we address the various issues in the design and implementation of the Internet Protocol (IP) in the evolving ATM local area networks.

DIPAK GHOSAL is a member of technical staff at Bell Communications Research.

H. JONATHAN CHAO is associate professor of electrical engineering at Polytechnic University.

DEBANJAN SAHA is a Ph.D. student in the department of Computer Science at the University of Maryland.

SATISH K. TRIPATHI is a professor and chair of the department of Computer Science at the University of Maryland, College Park.

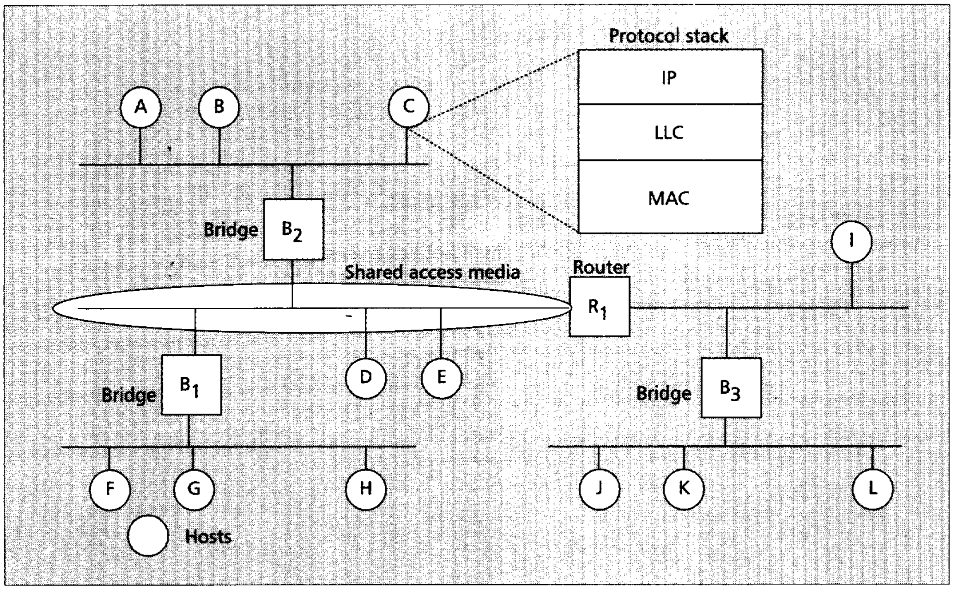
IP and IEEE 802 LANs

The Internet is a collection of LANs/MANs interconnected by routers and gateways. Each host in the Internet has a network layer address, referred to as the IP address. The address space is hierarchically organized and is used by the IP to route packets in the Internet. IP is the network layer protocol which provides a connectionless delivery mechanism [1]. Connectionless implies that there is no fixed logical and/or physical path from the source host to the destination host — each packet is routed independently and may potentially follow different paths in the network between the same source-destination pair. The basic unit of transfer, called a datagram, is divided into a header area and a data area. The header contains the source and destination IP addresses and the information for interpreting the data area.

Most existing LANs are based on shared media interconnects and employ the IEEE 802 family of LAN protocols, which includes the Ethernet, the Token Ring, and the Token Bus [2]. A LAN protocol defines both the physical and the data link layer protocols in the ISO reference model. In the IEEE 802 model, the data link layer protocol is divided into two layers: 1) the medium access control (MAC) layer which defines the mechanisms that are used to access, share, and manage the communication medium, and 2) the logical link control (LLC) that defines a common interface for different network layer protocols to inter-work with different MAC protocols.

Figure 1 shows an example of a LAN. Each host attached to a LAN has a globally unique MAC address which is six bytes in length and has a flat address space. In order for a host to send data to another host, it is necessary for the source host to know the MAC address of the destination host if they are in the same LAN or the MAC address of the next hop router if they are in different LANs. Thus, for example in Fig. 1, if A wants to send data to H it must know H's MAC address, whereas if A wants to send data to J, it must know router Rl's MAC address, which in turn must know J's MAC address. Resolving IP address to MAC address is done by an address resolution protocol, or ARP. For example, if A wants to send data to B and does not know B's MAC address, it broadcasts an ARP query with its own IP and MAC addresses and B's IP address. B responds to the query by sending an ARP reply with its own MAC address. When A receives the ARP response, it binds B's MAC and IP addresses and maintains it in a cache referred to as the ARP cache. Subsequent data transfer from A to B is done using B's MAC address obtained from the ARP cache.

Bridges are used to join different LAN segments to create a larger single LAN [2, 3]. In their simplest form, a bridge listens to a transmis-



■ **Figure 1.** *An example of a local area network.*

sion on one LAN segment and transmits it to all the other LAN segments to which it is connected. As specified in IEEE 802.1 [2], bridges are required to be able to learn to associate MAC addresses to specific bridge ports. For example, after listening to the various packets on the LAN in Fig. 1, bridge B2 can learn that the MAC addresses of hosts A, B, and C are associated with one of its ports while the MAC addresses of hosts D, E, F, G, H, and router Rl are associated with the other port. Bridges are transparent to the hosts. Thus when A sends an ARP query for F, it receives F's MAC address.

IEEE 802 LANs are based on shared-media interconnects. This implies that all communication is broadcast. This feature is effectively used by the address resolution protocol and the various connectionless datagram services.

ATM Networks

In contrast to IEEE 802 LANs, ATM networks are inherently connection-oriented. A connection must first be established between two ATM hosts before any data can be transferred [4-6]. Once the connection or the virtual channel (VC) is established between two nodes, data is transferred in fixed-size ATM cells. A virtual channel is identified by a virtual channel identifier or VCI. If a VC spans multiple links, each link can potentially have a different VCI for the VC. Each ATM cell carries the VCI of the connection it belongs to in the cell header. In the network, switches appropriately change the VCI values and route the cell from an input link to the appropriate output link.

The architecture of the ATM switch is shown in Fig. 2. Each input and output port has a controller, referred to as an input port controller (IPC) and an output port controller (OPC), respectively. A functional block diagram of the IPC is also shown in Fig. 2. It consists of a table, referred to as the VCI Table, which maps an input VCI (VCI^I) to an output VCI (VCI^O) and an output port address. Before the cells are released to the switching fabric, the VCI is replaced by the output VCI and the output port address is appended for self-routing. Each ATM switch has a switch controller that performs different switch management functions, including updating the table in the IPCs.

Each ATM host can be assigned an ATM address that could be based either on a hierarchical 8-byte-long ISDN telephone number scheme E.164 or a 20-byte address proposed by the ATM Forum [7]. The latter is modeled after the address format of an OSI network service access point. As shown in Fig. 3, it allows three formats which are discriminated using the AFI field. The 6-byte MAC address can be included in the ESI field.

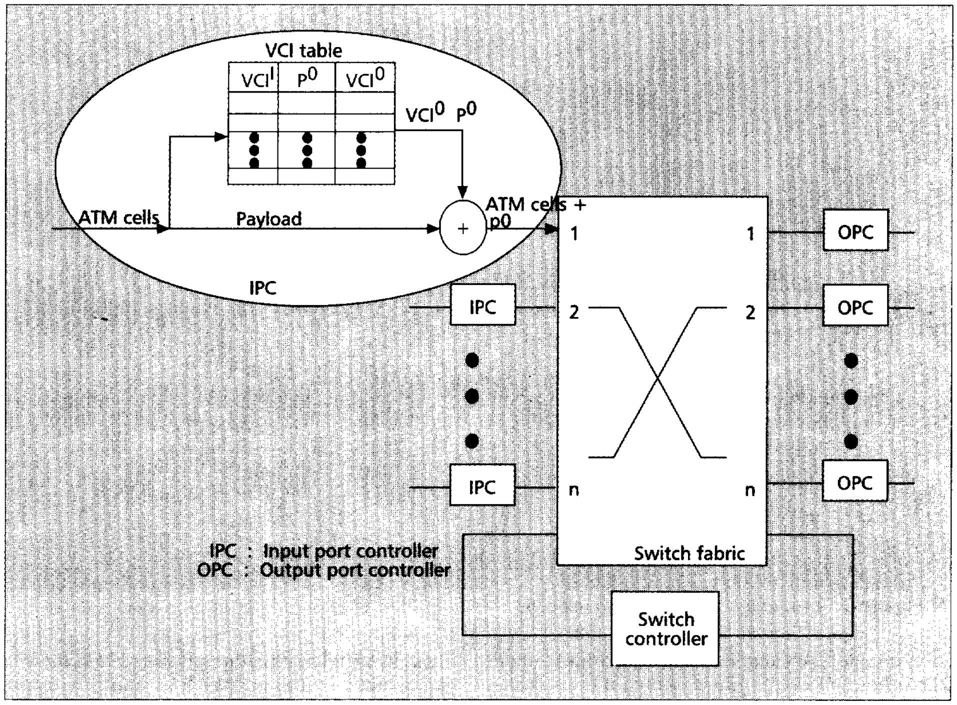
With respect to IP and other network layer protocols, ATM can be configured either as a separate data link layer protocol or as a MAC protocol below the LLC. The former approach results in IP and other network layer protocols to be implemented directly over ATM [8] (discussed in the section "IP Over ATM"). The latter approach is the key idea behind LAN emulation, and allows ATM switches to be transparently interconnected with shared-media legacy LANs running the IEEE 802 family of LAN protocols. The LAN Emulation Sub-working Group of the ATM Forum is actively involved in standardizing this scheme [9]. We discuss the various architectural issues of LAN emulation in the following section.

LAN Emulation

LAN emulation simply means that the point-to-point ATM switch should give the appearance of a virtual shared medium. From the point of view of the protocol stack, the ATM layer should behave like yet another IEEE 802 MAC protocol below the LLC [10]. The key attribute of all shared-medium interconnects is that all communication is broadcast, implying that all stations in a LAN receive all the packets, and they filter out the packets that they want to receive. Although ATM is connection-oriented, the broadcast feature can be emulated

Most existing LANs are based on shared media interconnects and employ the IEEE 802 family of LAN protocols which include the Ethernet, the Token Ring, and the Token Bus.

LAN *emulation simply means that the point-to-point ATM switch should give the appearance of a virtual shared medium.*



■ **Figure 2.** *ATM switch model.*

in an ATM network using dedicated servers.

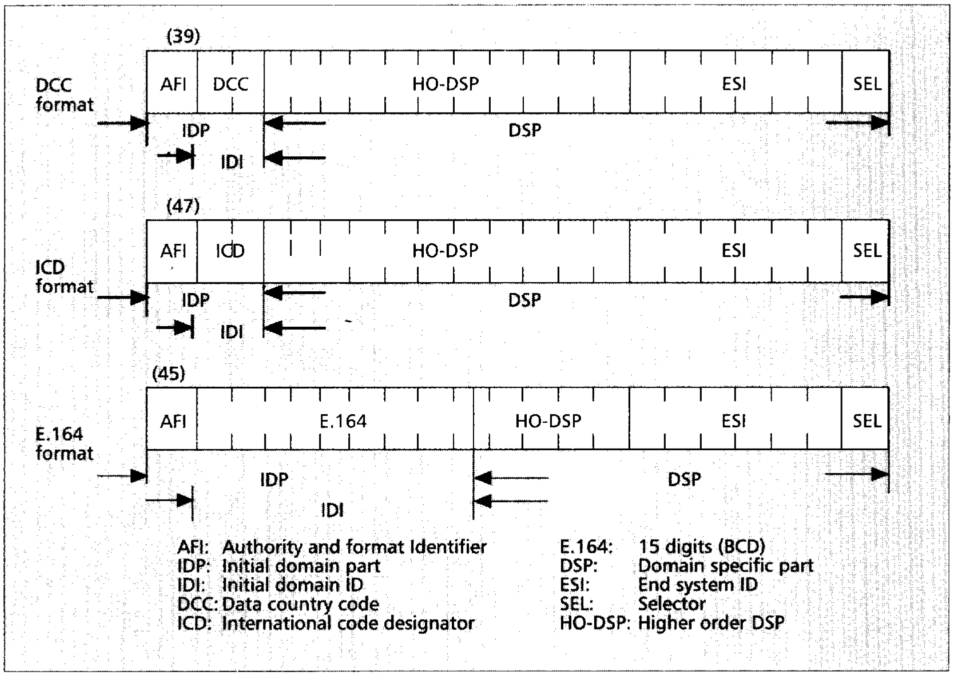
The LAN Emulation Sub-working Group of the ATM Forum has identified a number of different servers which include the LAN emulation (LE) server and one or more multicast servers such as the broadcast and unknown server (BUS), the specific multicast server (SMS), and the general multicast server (GMS) [9]. The LE Server provides functionality for registering and resolving MAC addresses and/or route descriptors to ATM addresses. The Multicast Servers are required to provide the connectionless data delivery characteristics of a shared network to hosts that are directly connected to the ATM network, referred to as the LE Clients. Since the number and the functionalities of these servers are not yet standardized, we only consider a single server which we refer to as the multicast server and identify its functional requirements.

One key function that the multicast server should support is the ability to multicast a MAC frame to a set of LE Clients. Given this functionality, LAN emulation can be simply supported by requiring all LE Clients to send all the MAC frames and ARP queries and replies to the multicast server, which can broadcast them to the LE Clients. In this approach, the LE Clients are responsible for selectively filtering out the packets they want to receive. Consider the LAN in Fig. 4 that is obtained from Fig. 1 by replacing an IEEE 802 LAN by an ATM switch. For simplicity, we assume that the multicast server is implemented as a central server and is placed on a globally known virtual channel. This is in contrast to a distributed approach in which the functionality can be distributed among multiple servers. Let us consider some specific examples of data transfers.

If A wants to send an IP packet to B, only the IEEE 802 LAN segments get involved. If A wants to send an IP packet to D and does not know D's MAC address, it first broadcasts an ARP request. The ARP request is received by bridge B2. Note that learning bridges bind MAC addresses to ports. Since the ARP query has no destination MAC address, the bridge simply copies the ARP query to all the valid outgoing ports which in this case is the port connected to the ATM switch. On the ATM side, the ARP query is received by the multicast server which broadcasts it to all the LE Clients. When D gets the ARP request, it replies by sending the ARP response to the multicast server which broadcasts it to all the LE Clients. Bridge B2 forwards the ARP reply to A and binds D's MAC address to the appropriate port. When A sends the IP packet in a MAC frame with D's MAC address, bridge B2 receives the frame and forwards it to the multicast server. The multicast server broadcasts it to all the LE Clients.

From the above discussion it is clear that in this scheme the multicast server is likely to become the bottleneck. Furthermore, this scheme does not utilize the advantage of point-to-point connection that can be setup across the ATM switch. In the IEEE 802 family of LAN protocols, it is the ARP function that crucially depends on the inherent broadcast feature of the shared medium interconnect. In view of this, it should be possible to architect the ATM switch and the LE Client interfaces in such a manner that only the ARP queries are processed by multicast server. All other data transfers use the point-to-point connections established across the ATM switch. Two key functions are required for this purpose: 1) the ability to resolve MAC addresses to ATM addresses, referred to as LE-ARP; and 2) the ability to perform connection management. Figure 5 shows the functional block diagram of an ATM host interface.

Translating MAC addresses to ATM address-

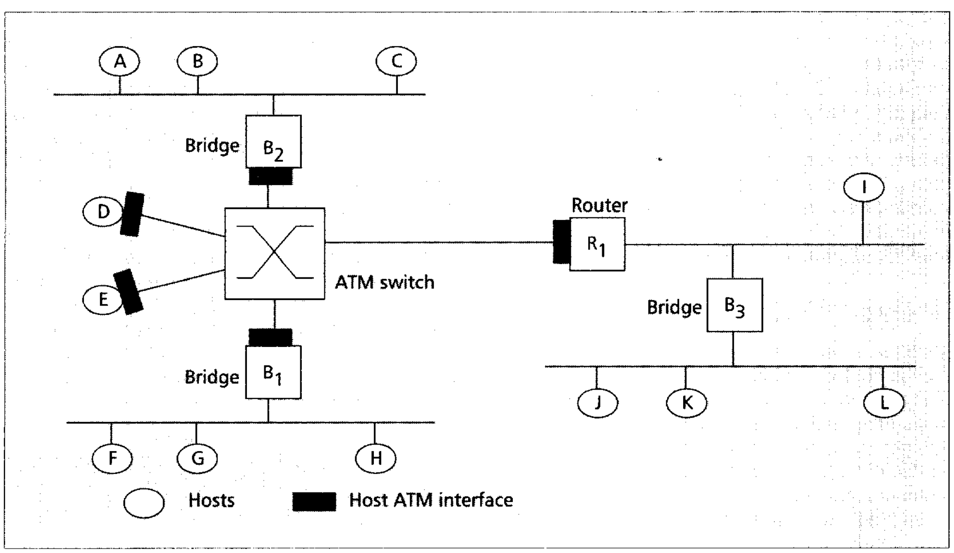


■ **Figure 3.** *Network service access point (NSAP) address format.*

LAN *emulation architecture must support all the well established network layer protocols efficiently.*

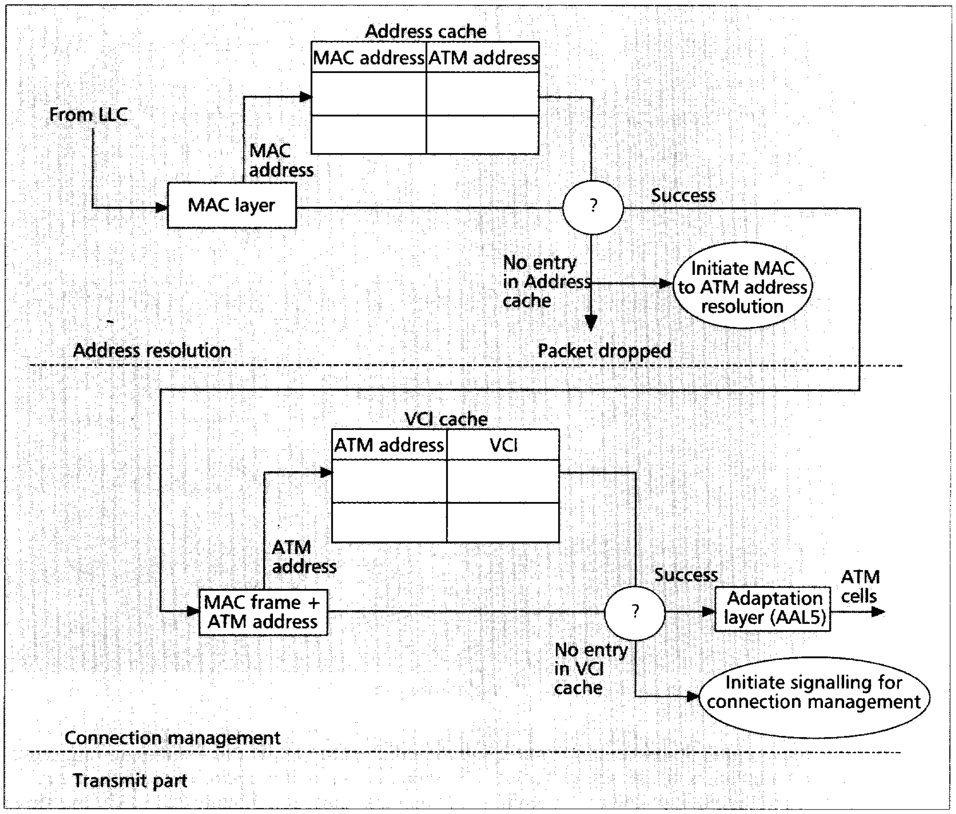
es can be done using either a broadcast approach or a server-based approach. In the server-based approach, an LE-ARP Server is placed on a globally known virtual channel. All MAC-to-ATM address resolution requests are sent to the LE-ARP Server, which can respond to the requester using a predetermined virtual channel. The interaction between the LE-ARP Server and the LE Clients can be implemented using a set of simple query response/messages. The broadcast-based approach relies on the switch broadcast capability.

Based on the above discussion, the data transfer from host A to host D in the previous example proceeds as follows: A broadcasts an ARP request for D which is received by bridge B2. The bridge sends the ARP query to all the valid outgoing ports, which in this case is the port connected to the ATM switch and more specifically that of the multicast server. The server broadcasts the ARP query to all the LE Clients. When D gets the ARP request, it responds by trying to establish a point-to-point connection to A. In the LE-ARP server, A's MAC address is associated with bridge B2's ATM address. So D establishes a point-to-point connection to bridge B2 and sends the ARP reply. Bridge B2 forwards the ARP reply to A and binds D's MAC address to the appropriate port. When A sends an IP packet in a MAC frame with D's MAC address, bridge B2 receives the frame and forwards it to the appropriate port. In this case the bridge uses



■ **Figure 4.** *A local ATM network.*

Since LAN Emulation defines the ATM layer as a MAC protocol below the LLC, supporting IP over an emulated LAN is the same as supporting IP over any IEEE 802 LAN.



■ **Figure 5.** *Host ATM interface.*

the point-to-point connection to D. This requires translation of D's MAC address to D's ATM address from the LE-ARP Server. This is followed by the connection management function, which is discussed in the section on connection management.

Since LAN Emulation defines the ATM layer as a MAC protocol below the LLC, supporting IP over an emulated LAN is the same as supporting IP over any IEEE 802 LAN. The key issue is to define a LAN emulation architecture that is both scalable and flexible in the sense that it can support the majority of the well established network layer protocols efficiently. Also, the architecture should be modular and easily extendible to the case when the the ATM sub-network consists of more than one ATM switch. Finally, the LAN emulation architecture should be able to handle not only unicast data transfer, but also broadcast and multicast data transfers. These could be implemented using special servers similar to the multicast server described above and are currently being standardized by the LAN Emulation Sub-working Group [9].

IP Over ATM

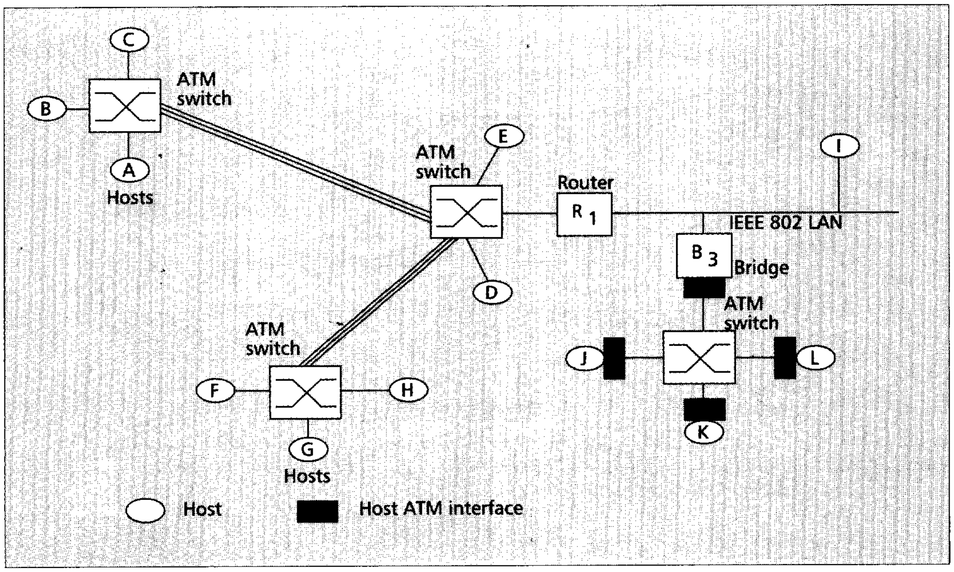
In a LAN consisting of only ATM switches, it is possible to simplify the protocol stack and run IP directly over ATM. Such ATM LANs are gaining popularity and are already being deployed [11]. The penetration of ATM LANs in the existing networks will primarily depend on how the cost performance ratio of ATM compares with competing technologies such as 100 Mb/s Ethernets and FDDI rings. With the current interest in ATM technology, it is likely that ATM switches and interfaces will become faster and cheaper than the competing shared medium technologies. In view of this, we envision the emergence of homogeneous ATM sub-networks interconnected to homogeneous/heterogeneous sub-networks via routers. Figure 6 shows such an evolution of the ATM LAN in Fig. 4.

Implementing IP directly over ATM will require translating an IP address to an ATM address [8]. The straight forward approach is to maintain a server, referred to as the IP-ATM-ARP Server, on a globally known VC in the ATM sub-network. The IP-ATM-ARP Server needs to maintain tables that can translate an IP address to an ATM address. The interaction between the hosts and the IP-ATM-ARP Server can be implemented using a simple query/response protocol. Table 1 lists a minimal set of messages that will be required for a simple implementation.

The host interface is the same as that in the case of the LAN Emulation except that the address cache now contains translations of IP addresses to ATM addresses. When the host needs to send a packet to a destination IP host, it obtains the corresponding ATM address from the address cache and passes the IP packet and the ATM address to the processing entity that performs the connection management function. If the address cache does not have an entry for the destination IP host, an ARP query is sent using the following message.

VCI used	Payload				
	Type	Field 1	Field 2	Field 3	Field 4
VCI-U	ARP query	IP_S	ATM_S	IP_D	?

Note that VCI-U is the virtual channel identifier for the globally known VC to the IP-ATM-ARP



■ **Figure 6.** *Interconnecting homogeneous and heterogeneous ATM sub-networks.*

Implementing IP over ATM will require translating IP addresses directly to ATM addresses.

Server. On receiving this message the IP-ATM-ARP Server will return the destination ATM address to the source using the following ARP Reply message

VCI Used	Payload				
	Type	Field 1	Field 2	Field 3	Field 4
VCI-ATM-S	ARP reply	IP_S	ATM_S	IP_D	ATM_D

where the VCI used is obtained from the reserved VCI table in the IP-ATM-ARP Server.

Address resolution is followed by connection set up and data transfer. If the VCI cache has a VCI for the destination ATM address, the IP packet along with the VCI is sent to the AAL5/ATM processor, which creates the ATM cells and passes them on to the port controller. Note that the ATM layer needs to provide interfaces to different network layer protocols. There are two approaches to multiplexing network interconnect traffic over ATM [12]. The first method allows multiplexing of multiple protocols over a single ATM virtual circuit. The protocol of a carried PDU is identified by prefixing the PDU by an IEEE 802.2 LLC header. The second method does higher-layer protocol multiplexing implicitly by using different VCs for different protocols. It is envisioned that VC-based multiplexing will be dominant in environments where dynamic creation of large numbers of ATM VCs is fast and economical. LLC Encapsulation, on the other hand, may be more suitable if the ATM network only supports (semi-) permanent virtual circuits (PVCs).

When a new host is added to the network, it goes through a registration process with the ATM network and obtains the ATM address. In order to update the IP-ATM-ARP Server it sends an `add host` message giving its IP and ATM addresses. The IP-ATM-ARP Server updates the IP to ATM address map and allocates a new reserved VCI to the new host and returns a `confirm` message with the new reserved VCI. The host can use this VCI to discriminate between messages received from the IP-ATM-ARP Server and other hosts. The above process can also be adopted to allow port mobility of hosts. The address and the VCI caches may be invalidated periodically and and rebuilt just as in the case of a new host.

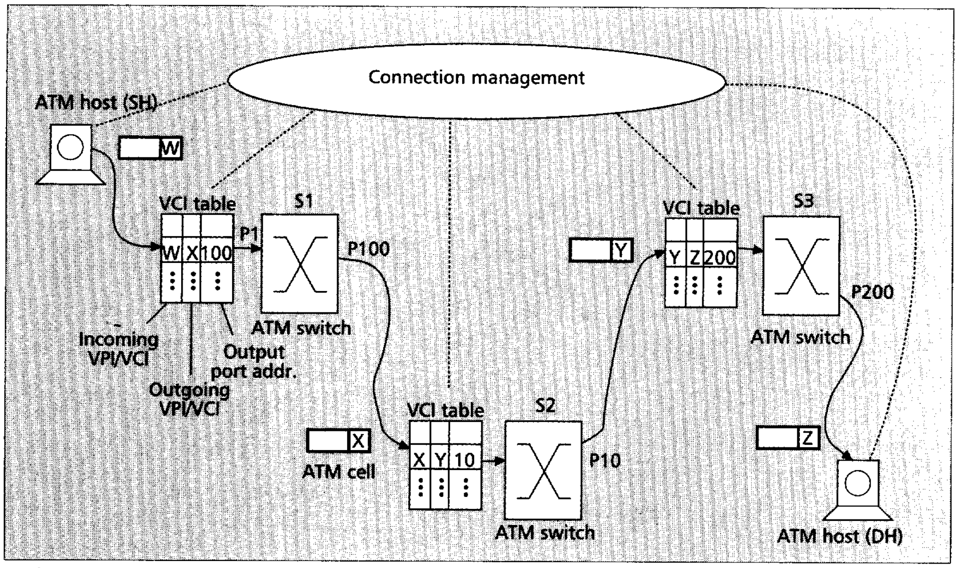
Connection Management

The task associated with connection management is the same irrespective of whether ATM is configured as a complete data link layer protocol supporting IP or as an IEEE MAC protocol below LLC, as in the case of LAN Emulation. Both cases illustrate the various issues that are inherent in supporting a connectionless service over a

Type	Field 1	Field 2	Field 3	Field 4	Comment
ARP query	Source IP address	Source ATM address	Destination IP address		From IP host to server to obtain destination ATM address.
ARP reply	Source IP address	Source ATM address	Destination IP address	Destination ATM address	From server to IP host with destination ATM address.
Add host	IP address	ATM address			From IP host to server to update ATM address.
Confirm	IP address	ATM address	Reserved VCI		From server to IP host with reserved VCI.

■ **Table 1.** *Signaling messages between host and IP-ATM-ARP Server.*

The task associated with connection management is the same irrespective of whether ATM is configured as a complete data link layer protocol supporting IP or as an IEEE MAC protocol below LLC.



■ **Figure 7.** *Connection management functions.*

connection-oriented link layer protocol. There are two primary approaches in performing the connection management functions — one based on permanent virtual channels (PVCs) and the other based on switched virtual channels (SVCs). In the SVC based approach, each ATM host must he capable of running the signaling protocol such as, Q.2931 [7], to establish and tear down point-to-point and point-to-multipoint connections.

Let us consider in detail the task of establishing a connection between source host (SH) and destination host (DH) in Fig. 7 using Q.2931. Connection management is initiated with ATM host SH sending a SETUP message to switch Sl which processes the request and if the call is accepted sends a CALL PROCEEDING message with VPI/VCI to SH and a SETUP message to switch S2. Switch S2 in turn processes the request and sends a CALL PROCEEDING message back to Sl with the VPI/VCI and a SETUP message to S3. Switch S3 responds with a CALL PROCEEDING message to S2 and a SETUP message to the destination host. If the destination accepts the call request, it updates its receive VCI cache and sends a CONNECT message back to S3, which updates the VCI Table (Fig. 2) in the corresponding input port controller and sends a CONNECT message back to S2. The CONNECT message eventually propagates back to the SH, which then updates the VCI cache and data transfer follows. Tearing down the call is also done hop-by-hop using RELEASE and RELEASE COMPLETE messages.

The SETUP message that establishes an ATM connection to be used for IP contains the AAL parameters, ATM traffic descriptor and QoS parameter along with other information. The AAL parameters to be specified include AAL type and maximum service data unit size. The ATM traffic descriptor characterizes an ATM connection in terms of peak cell rate (PCR), sustainable cell rate (SCR), and maximum burst size. This information is used to allocate resources in the network. In general the ATM traffic descriptor for a connection carrying IP traffic will be determined by several factors, such as application requirements, cost of service, etc. The QoS classes most applicable to IP connections are Class 0 (unspecified QoS class), Class 3 (specified QoS class for connection oriented data transfer) and Class 4 (specified QoS class for connectionless data transfer). The available class of QoS parameters and ATM traffic descriptors is specific to the ATM network. It is the responsibility of the source host to choose the QoS class that is offered by the network and satisfies its requirements.

In the PVC-based approach, PVCs are established *a priori* between every pair of ATM hosts and maintained in a central table that can be updated when new hosts join the network. Connection management involves simply a table lookup to determine the VCI to be used depending on the destination ATM address. For small ATM networks with few hosts, such a scheme is straightforward and easy to implement. However, it is not scalable to networks with a large number of hosts.

A modification to the above PVC-based approach is to maintain PVCs between ATM ports and multiplex them among the hosts attached to the port. In this approach, a number of PVCs are allocated for IP services between each pair of ports [13]. A central entity called the Connection Manager allocates and de-allocates PVCs to different connections. The connection manager maintains the following tables:

- Free list: this is a list of unallocated VCs between each pair of input and output ports.
- Allocated list: this is a table of ATM source/destination addresses and the corresponding allocated VCs.

A minimal set of query-response messages exchanged between the connection manager and the end-hosts is listed in Table 2. Let us consider how a connection is set up between SH and DH using the connection manager. If the VCI Cache in SH has no entry corresponding to DH, the connection management function is initiated. This results in a query message to be sent to the connection manager. If there is an entry in the allocated list for the particular source destination pair, then the connection manager responds to SH with the reply message including the input VCI and sends an update message to DH with the output VCI, which results in the receive and the transmit VCI caches in the DH, respectively, and the SH to be updated and data transfer to be initiated.

Type	Field 1	Field 2	Field 3	Comment
Query	Source ATM address	Destination ATM address	-	From ATM host to CM to obtain VCI for a destination ATM address.
Reply	Source ATM address	Destination ATM address	VCI^{I}	From CM to Source ATM host with a new transmit VCI.
Update	Source ATM address	Destination ATM address	VCI^{O}	From CM to destination ATM with a new receive VCI.

■ **Table 2.** *Messages exchanged between an ATM host and the Connection Manager.*

If there is no entry in the allocated list then the connection manager assigns a new VC from the free list for the appropriate port pairs and follow the same procedure as above. If the free list is empty, the connection management protocol fails and a higher level network management protocol is initiated to re-map the allocated VCs and/or allocate new VCs between the port pairs.

An important aspect of connection management is the maintenance of the VCI cache. It is important to revalidate entries after they remain in the VCI cache for a fixed time. Each cache entry has a time-to-live field associated with it. When a new entry is added or accessed (to resolve an address to a VCI) the time-to-live field is initialized. As time proceeds, the cache manager decrements the time to live field, and discards the entry when the value reaches zero. It also initiates a call release procedure.

Concluding Remarks

In this article we have addressed the issues that are involved in implementing IP in ATM LANs. In LAN Emulation, ATM is configured as an IEEE 802 MAC protocol below LLC. This not only allows ATM switches to be transparently integrated with the IEEE 802 family of LAN protocols but also allows ATM to be transparent to IP and other network layer protocols. We addressed the impact of IP on the design of a scalable and an efficient LAN Emulation architecture. If ATM is configured as a separate data link layer protocol, then IP must be implemented directly over ATM. The key issue in this case involves resolving IP addresses to ATM addresses and providing transparency of ATM to different network layer protocols. We also investigated the different approaches to connection management and described a PVC-based approach that is both efficient and scalable. There are other issues which need to be resolved before finalizing the standards. Routing across multiple logical IP subnets, support for IP multicasting, and many other performance and reliability issues are being actively pursued in the ATM Forum, IETF, and other standard bodies. Finally, it should be pointed out that ATM can provide the full functionalities of the network layer and data link layer protocols, and as a result, it is possible to implement transport layer protocols such as TCP directly over ATM. Although this may result in a very efficient protocol stack, the forces to inter-work ATM with the huge base of IEEE 802 family of LAN protocols and network layer protocols such as IP is likely to limit its penetration.

ATM can provide the full functionalities of the network layer and data link layer protocols.

References

[1] D. E. Comer, Internetworking With TCP/IP, Volume 1, Principle, Protocols, and Architectures, (Prentice-Hall, 1991).
[2] R. Perlman, Interconnections: Bridges and Routers, (Addison-Wesley, 1992).
[3] G. H. Clapp, "LAN interconnection across SMDS," IEEE Network, vol. 5, no. 5, Sept., 1991, pp. 25-32.
[4] J.-Y. Le Boudec, "The Asynchronous Transfer Mode: A Tutorial," Computer Network and ISDN System, vol. 24, 1992, pp. 279-309.
[5] Telco Systems, Asynchronous Transfer Mode: Bandwidth for the Future, Telco Systems 1992.
[6] Hughes LAN Systems, ATM in Private Networking, INTEROP '93 Version.
[7] ATM Forum, ATM User-Network Interface Specification Version 3.0, (Prentice Hall, 1993).
[8] M. Laubach, Classical IP and ARP over ATM, RFC 1577, Information Sciences Institute, Jan. 1994.
[9] LAN Emulation SWG Drafting Group, "LAN Emulation Over ATM: Draft Specification - Revision 2," ATM Forum/94-0035R2+, April 15, 1994.
[10] P. Newman, "ATM Local Area Networks," *IEEE Commun. Mag.*, vol. 32, no. 3, March 1994, pp. 86-98.
[11] E. Biagoni, E. Cooper, and R. Sansom, "Designing a Practical ATM LAN," *IEEE Network*, vol. 7, no. 2, March 1993, pp. 32-39.
[12] J. Heinanen, Multiprotocol Encapsulation over ATM Adaptation Layer 5, RFC 1483, July 1993.
[13] D. Saha, D. Ghosal and J. Chao, "A Design for Implementation of Internet Protocol in a Local ATM Network," International Communication Conference, New Orleans, LA, 1994.

Biographies

DIPAK GHOSAL is a member of technical staff in Bell Communications Research, Red Bank, New Jersey. His current research interests are in the areas of parallel protocol processing architectures for high-speed networks, robust network architectures for mobile communication systems and performance evaluation of computer and communication systems.

H. JONATHAN CHAO received his Ph.D. degree in electrical engineering from The Ohio State University, in 1985. He is associate professor of electrical engineering at Polytechnic University, Brooklyn, NY, which he joined in January 1992. His areas of research include large-scale multicast ATM switches, photonic ATM switches, high-speed computer communications, and congestion/flow control in ATM networks. He holds more than a dozen patents and has published more than 40 journal and conference papers in the above areas. From 1985 to 1991, he was a member of technical staff at Bellcore, doing research in SONET/ATM-based broadband networks.

DEBANJAN SAHA received B.Tech. in computer science and engineering from The Indian Institute of Technology, Kharagpur, in June 1990. He is currently a Ph.D. student in the Department of Computer Science at the University of Maryland, College Park. His doctoral research investigates protocol issues in ATM networks.

SATISH K. TRIPATHI is a professor and chair of the department of Computer Science at the University of Maryland, College Park, where he has been on the faculty since 1978. He attended the Banaras Hindu University, the Indian Statistical Institute, the University of Alberta, and the University of Toronto. He received his Ph.D. in computer science from the University of Toronto. For the last 15 years he has been actively involved in research related to performance evaluation, networks, real-time systems and fault tolerance. He has served as the member of the Program Committee and program chair for various international conferences. He has guest edited special issues of many journals and serves on the editorial board of Theoretical Computer Science.