Tutorial

Getting Started with ATM

Hewlett-Packard Company
Enterprise Networking and Security Division
Steve Hampton, Marie-Noelle Jeanson, Bernard Virondeau, Vithoune Vongphacdy

0. Abstract

ATM has grown out of a need for a high speed networking technology capable of efficiently supporting multimedia traffic, even over large distances. The breakdown of data into small cells makes this possible, and allows each connection to specify its Quality of Service requirements to the network.

Although users can set-up PVC connections themselves, signalling allows them to let the network set-up and manage SVCs dynamically. In order to agree upon a certain Quality of Service at connection set-up, an end-user will choose a Service Category (CBR, VBR, UBR or ABR) and optionally, the attribute values (e.g. value of Cell Loss Ratio). Mechanisms such as ABR flow control can even allow certain attribute values to change during the connection.

Although native ATM applications and services can provide the full benefits of ATM, the focus today is on coexistence with existing data applications and networks: Classical IP allows ATM hosts to run most IP applications. LAN Emulation allows ATM hosts to behave like legacy LAN clients, run the same applications and efficiently communicate with them. MPOA allows switched ATM backbones to benefit from LAN Emulation switching, even across subnet boundaries.

1. Introduction: Why ATM ?

Application-driven: The evolution of current applications (e.g. those based on the IP protocol), and the advent of data-intensive and multimedia applications (e.g. high resolution video broadcasting, distance learning or medical imaging), have created a need for a flexible and efficient network technology capable of integrating data, voice and/or video over the same high speed infrastructure.

Network-driven: Moreover, the sheer growth in the number of global networked users (10-fold expected growth of Internet users in the next 5 years) and in the amount of information and resources being shared over private networks (Intranets), ask for a flexible and scalable infrastructure, capable of adjusting to fast changing network traffic profiles.

Which alternative? Existing high speed networking technologies (FDDI, Fast/Gigabit Ethernet, Frame Relay) are not as well suited as **ATM (Asynchronous Transfer Mode)**, for these new requirements. The fundamental purpose of ATM is to provide each application with the network resources most adapted to its needs, in terms of **Bandwidth** and **Quality of Service**. ATM was also designed as an integrated and scalable means of interconnecting existing **LAN and WAN** networks such as Ethernet, FDDI, Frame Relay or ISDN. ATM is a result of an international **Standards** effort and has already been endorsed by major equipment vendors and network service providers.

- Figure 1: High Speed Networking Alternatives -

2. Basic Concepts

Let's see how ATM manages to provide these benefits.

The Cell: the concepts of ATM are based on a small, fixed length (53 Bytes) data unit: the cell. The cell is the smallest common denominator capable of transporting data, voice and video traffic efficiently. With ATM, user data is segmented into such cells.

Bandwidth and Scalability: The fixed length cell allows for hardware optimization: **switching** can take place at hardware speeds with minimal processing overhead because routing information is coded within each cell at a fixed place.

Multimedia Support: The small cell size allows for efficient **multiplexing** of various application traffic over the same physical link, whether these applications require high bandwidth (e.g. data), or low transfer delay and jitter (e.g. real-time voice and video), and even over very long distances.

- Figure 2: The Cell across an ATM network -

LAN/WAN Integration: ATM networks have a star topology. ATM switches can be used to interconnect LAN end-systems (e.g. PCs) or network devices (e.g. other switches or routers) hosting ATM adapters or ports. ATM can also be used for LAN-to-LAN interconnection of distant ATM, FDDI, Ethernet, or Token Ring networks. Due to its efficient cell switching, ATM is the only technology which can carry multimedia traffic across both LAN and WAN networks today. Interconnection mechanisms with existing networks (e.g. Ethernet LAN Emulation) are aimed at preserving current network investments and allowing a smooth transition to next generation network infrastructures.

An ATM call: Very much like today's telephone and unlike Ethernet LANs, ATM is connection oriented; an ATM number must be "dialed" and a connection must be established before end users can start exchanging information cells.

The ATM address: This 40 digit long address is composed of a Network Prefix obtained from the ATM network (i.e. area code), and an End System Identifier which is retrieved from the ATM interface card hardware and which uniquely identifies the host, as far as the ATM network is concerned (i.e. telephone extension number or Ethernet MAC address).

Quality of Service: At connection set-up, a caller will negotiate the level of quality for that connection with the network. This Quality of Service (QoS) is attributed on a per-call basis and guarantees specific traffic parameter values for the connection (e.g. bandwidth, delay or jitter).

The ATM Protocol Stack: once a connection has been established and QoS has been agreed upon, end-users can start transmitting data by coding and decoding it across the three ATM protocol layers:

The Adaptation layer (**AAL**) will adapt higher layer application data to ATM's cell format by segmenting it into 48 Byte cell payloads (and vice-versa).
The **ATM** layer will generate the cells by adding the 5 Byte cell header (which contains routing and QoS information). It will therefore be able to handle cell routing and flow control across the network.
The Physical layer (**PHY**) will prepare the cells for binary transmission and reception over the physical medium (e.g. twisted pairs, fiber optics); it will generate line coding, bit timing and HEC. SONET and SDH are the standards most commonly used for the PHY.

- Figure 3: The ATM Protocol Stack -

Standards: ATM has been standardized to facilitate interoperability between different vendor solutions. Since the definition of the initial ATM specifications, the ITU-T, ATM Forum and IETF have been working on extended functionality in order to better respond to applications needs (see Appendix A for detail).

ATM can be summarized as a "Fast switching technology based on short fixed length packets". Let's take a closer look at the way ATM actually works.

3. ATM Switching

The ATM User-Network Interface (UNI) defines how an end-user interacts with the ATM network, in order to set-up **ATM Connections** dynamically (UNI Signalling)**.**

Before exchanging data cells, end-users must establish a connection. This end-to-end connection requires setting up intermediate **Virtual Circuits (VCs)** between all ATM switches along the way. Virtual Circuits can also be bundled together into a **Virtual Path (VP)** for common management. **VPI/VCI** Identifiers inserted in the ATM cell header will uniquely identify the connection at each end, and between two ATM switches. These switches will rely on their "VPI/VCI-to-physical port" mapping tables to update the cells VPI/VCI values and route them along the appropriate physical path.

- Figure 4: The ATM Switching Mechanism -

There are two ways of setting up an ATM connection:

PVCs: Some applications may require an ATM connection to be set-up once and for all. In that case, a user can manually define static Permanent Virtual Circuits (or PVCs) from one host, within all intermediate mapping switches tables, to the other host. No signalling mechanism is needed.

SVCs: Most applications can't be bothered managing ATM connections themselves. They would rather use an automated mechanism capable of setting-up ATM connections in a split-second, using more flexible and reliable means. This is the role of ATM Signalling i.e. the dynamic establishment, maintenance and clearance of ATM connections (Switched Virtual Circuits or SVCs). The ATM Signalling protocol manages the attribution of VPI/VCI numbers for each segment of the connection, and their mapping with the hosts' ATM addresses at each end point of

the connection. Signalling messages travel between two ATM nodes across a pre-defined VPI/VCI (0, 5) . These nodes can be a user and its ATM network switch, (User-Network Interface requiring UNI Signalling [2][3]) or two ATM switches (Network-Network Interface requiring Public or Private NNI Signalling [4]).

ILMI: ATM UNI Signalling requires end users to exchange certain parameters with the network (e.g. their own configuration and their 40 digit long ATM address which uniquely identifies them). The Integrated Local Management Interface **(ILMI)** protocol [5] uses SNMP to retrieve those parameters from the ATM device's Management Information Base (MIB), and supply them to the network, for connection set-up and management without requiring user interaction. ILMI messages travel across another pre-defined VPI/VCI (0, 16).

An ATM connection can either be point-to-point or point-to-multipoint:

Point-to-Point connections can be used by applications such as file transfer or Client/Server Internet web browsing.

Point-to-Multipoint connections can be used to emulate Ethernet LAN broadcast mechanisms, or for applications such as video broadcast: this type of connection can be initiated by the root (e.g. TV broadcast company), or joined by end-users or leafs (e.g. TV viewers).

4. Quality of Service and Traffic Management

Quality of Service (QoS) is the level of quality guaranteed by the network for each ATM connection. It is managed by network **Traffic Management** [5] whose role is the prevention of congestion in the network as well as in the end-users, and the promotion of efficient use of network resources.

Why QoS?
QoS today: When you use the Internet today, the bandwidth you get randomly depends on the network load or the number of other "contending" users on each link along the path. There is no way for your application to request the network resources it needs (e.g. low delay or jitter to run real-time voice applications). Also, there is no mechanism for the network to dynamically manage its resources to avoid congestion and fairly allocate resources to applications in order of priority.

Application incentive: data, voice and video have very different characteristics and network requirements; e.g. high bandwidth and low loss for data, low bandwidth and low delay and jitter for voice. The goal of each application is to get the Quality of Service most appropriate to its need.

Network incentive: the network must commit on a certain Quality of Service at each connection set-up, despite other simultaneous connections. To do so, it must dynamically monitor its resources to avoid congestion and best utilize the links (e.g. not refuse a connection if it can be serviced).

What is QoS?
In order to simplify QoS negotiation and attribution, several ATM Service Categories have been pre-defined for generic application requirements [1][5]:

- Figure 5: ATM Service Categories -

CBR (Constant Bit Rate): This category is aimed mainly at **highly sensitive real-time** applications, it can also be used for any constant stream application requiring **circuit switching quality**. It guarantees **fixed dedicated resources** (e.g. bandwidth) that are statically reserved, whether they fully utilized by the user or not. It can be considered the most costly in terms of network resources.

rt-VBR (real time -Variable Bit Rate): This category is intended for the majority of **real-time** applications. These time-sensitive applications are immune to slight data loss, and rarely transmit useful data 100% of the time. For instance, silences in one phone conversation could be used by others to transmit useful data. **Statistically multiplexing** such applications would make a much more efficient use of network resources, instead of allocating fixed bandwidth CBR channels to each.

nrt-VBR (non real time -Variable Bit Rate): This category would be used by most **bursty data** applications, characterized by short requests/replies requiring very fast response time and high **reliability**. Such bursty traffic could also benefit from statistical multiplexing with minimal risk of congestion.

UBR (Unspecified Bit Rate): This category offers **no Quality of Service** guarantee whatsoever. It is aimed mainly at **low priority data** traffic which could then rely on higher protocol layers for service as it would on regular LANs (e.g. TCP/IP error checking, retransmission and flow control). Although it is the least demanding in terms of network resource requirements, a single cell discard due to congestion can trigger a full packet retransmission (up to 191 cells for a Classical IP packet) by higher layers and significantly impact the overall performance.

ABR (Available Bit Rate): This category offers low cell loss, and bandwidth-on-demand above a guaranteed minimum rate. It is the most effective way of running today's **data** applications over ATM while **avoiding congestion** in the network. ABR's main value lies in its elaborate flow control mechanism. It allows end-systems to adjust their transmission rate dynamically according to the network's available resources.

- Figure 6: ATM Service Category Description -

How does ABR flow control work?

Periodically, each ABR connection polls the network by sending a "Resource Management" **RM cell**. This cell is looped back at the destination. Network switches and the destination can use this RM cell to inform the source about their available resources and congestion status. Based upon the **feedback** it receives in the RM cells, the source can adjust its transmission rate and back off in case of congestion. In the forward direction, switches can imply to the destination that congestion is present by setting the Explicit

Forward Congestion Indication (**EFCI**) bit in data cell headers (1). The destination (2) or any switch in either direction can also mark the RM cells (3) to specify the "**Explicit Rate**" (modify rate to xx Mbps) or "**Relative Rate**" (congestion is pending) in order to suggest the rate to the source (4).

For more responsiveness to **congestion**, this feedback loop can be reduced to anywhere between the source and any intermediate switch acting as a "**Virtual Source/Virtual Destination**".

- Figure 7: ABR Flow Control Mechanism -

How does Traffic Management work?

When requesting an ATM connection, a **Traffic Contract** needs to be agreed upon between the network and the ATM end node. According to the type of traffic for that connection, the end node provides the network with its Quality of Service request. It specifies the **Traffic Category** (CBR, rt-VBR, nrt-VBR, UBR or ABR), and the **Traffic Parameter** values (e.g. Cell Loss Ratio). The network, in turn, allocates resources that match the parameters or, if the resources are unavailable, rejects the call. Once the call is accepted, it is the end node's responsibility to adhere to the traffic contract, as the network will discard any cells which are non-compliant.

- Figure 8: The ATM Traffic Contract -

Which Benefits?

For an application, it is the ability to specify and obtain the network resources it needs. For the network, it is the capability to control congestion and efficiently allocate its resources, as opposed to the best effort on today's LANs.

Which Limitations?

Most user implementations over ATM today and in the foreseeable future, are likely to be for legacy data network applications, using legacy protocols such as IP or interconnecting to legacy networks such as Ethernet LANs. These applications are not able to specify the Quality of Service they require, so until the next generation of native ATM applications, most will rely on external and less flexible processes (e.g. ATM adapter software driver) for QoS negotiation and configuration.

5. Native ATM Applications

Why Native ATM Applications?

ATM features enable a wealth of new possibilities to application developers. Data, voice and video can smoothly be integrated into ATM applications and directly tap into this technology's capabilities, such as the flexible set-up and management of connections and the broad range of QoS that can dynamically be negotiated and guaranteed from the network.

Unlike most applications based on APIs, the Native ATM Services Semantic [9] can also be used to provide operating system kernel interfaces between Native ATM Services and other ATM or non-ATM protocols such as LAN Emulation, IP, or X.25.

What is a Native ATM Application?

A Native ATM Application is simply an application that has been developed directly over the ATM Adaptation Layer, e.g. using ATM APIs. Applications which could readily benefit include multimedia applications, video-on-demand and broadcasting, medical imaging, and collaborative networked applications such as videoconferencing.

- Figure 9: The Native ATM Services Model -

Which Benefits?

A native ATM application can deliver the following benefits:

- **Performance:** e.g. making abstraction of protocol overheads and mechanisms (e.g. TCP/IP) which may not be needed.
- **QoS:** e.g. obtaining guaranteed service from the network, such as dedicated bandwidth and low delay (CBR) for real-time voice and video, bandwidth-on-demand for data (ABR).
- **Resource control and management**: e.g. the opening and closing of Switched or Permanent Virtual Circuits between hosts, with different characteristics closely mapped to their purpose; also the distribution of these connections to the correct application; finally the Local Resource Management providing ATM network management information using ILMI.

Which Limitations?

Portability: the main limitation is that these applications are specific to their environment (ATM); just like any other APIs, they cannot easily be ported on other network technologies. They only address particular ATM needs, and are not adapted to heterogeneous network environments comprising legacy Ethernet hosts for instance.

Consequently, several protocol suites have been developed, in order to interface ATM with today's most common network applications and environments:

- Classical IP over ATM: allowing ATM hosts to run IP-based applications.
- LAN Emulation and Multi Protocol Over ATM: allowing ATM hosts to communicate efficiently with non-ATM hosts and run most current network applications.

6. Classical IP over ATM

What is Classical IP ?

The majority of networked applications today are based on the IP protocol. These include Internet applications such as web browsers, NFS, or File Transfer Protocol.

In order to run these applications over ATM, some kind of protocol "glue" is needed between the application's IP layer (Internetwork Layer 3 in the OSI model) and the ATM protocol stack, (i.e. on top of ATM Adaptation Layer as described in Figure3). A protocol mechanism allowing IP applications to open point-to-point ATM connections is also needed.

"Classical IP over ATM" was defined for these purposes by the IETF [10][11][12] :

- It defines a mechanism for the transmission of IP traffic over ATM.
- It allows an ATM network to be split into Logical IP Subnets (LIS) in order to emulate current IP subnets.

The ATM Classical IP subnet is termed "logical" in the respect that the IP address and subnet assigned to a Classical IP host is purely arbitrary and does not depend on the physical location of the host. In a similar way as Ethernet Virtual LANs (IEEE802.1 VLANs), two hosts separated by distant switches could still belong to the same LIS; two hosts connected to the same switch could belong to different LIS and require a router to communicate.

- Figure 10: Logical IP Subnets (LIS) with Classical IP -

How Does Classical IP Work?

IP over Ethernet: Over Ethernet, hosts are identified according to their **MAC** address (OSI Layer2). Within an IP subnet, an IP application wishing to reach a remote host will need to figure out the remote hosts' MAC address, either in its host's cache table or by **broadcasting an IP_ARP** (Address Resolution Protocol) request to all hosts on the subnet. The remote host will recognize itself as the recipient of the ARP, and reply with its MAC address and the connection will be set.

IP over ATM (SVCs): Over ATM, hosts are identified according to their **ATM** address. Within a LIS, an IP application will need to figure out the remote host's ATM address in order to set-up an ATM connection with it. As opposed to Ethernet, ATM is a non-shared medium and does not support direct broadcasts, as needed for Ethernet-like ARP requests. Classical IP will use a device on the subnet called the **ATM ARP Server** which will maintain an IP-to-ATM cache table for all hosts on that subnet and answer their request for IP-to-ATM address resolution.

Classical IP connection set-up: In order to set up an ATM SVC connection, the source must first find out a remote host's ATM address. It will ask the ATM ARP server, by sending it an ATM_ARP Request (1), along with the remote host's IP address. The ATM ARP Server will look up its cache and answer with an ATM_ARP Reply (2) containing the remote host's ATM address. At this point, the initiator will be able to set-up an ATM SVC connection (3), e.g. using VPI/VCI (0,183).

- Figure 11: Setting up a Classical IP SVC Connection -

This process is entirely transparent to the user who only makes reference to the destination hostname or IP address, as simply as in any "classic" IP network.

In either case (Ethernet or ATM), if the remote host happens to be in a different subnet, an **IP router** will be needed to forward the request to other Ethernet IP subnets or ATM LIS.

IP over ATM (PVCs): this signalling mechanism is not necessary since PVCs are set-up manually. All that is required from a host at one end of a PVC, is to be able to supply its own IP address upon request from the host at the other end, using the Inverse ATM ARP protocol.

What is needed for Classical IP over ATM?

LIS: firstly, you will need to define how you wish to segment your ATM network, i.e. define all Logical IP subnets (LIS), as you would on any IP network. They would be separated by routers.

ATM ARP Server: then you will need to assign the ATM ARP Server task to one device in each LIS. The ARP Server functionality is available in most ATM adapters and ATM switch software.

ATM ARP Clients: you will need to configure all other hosts on the LIS subnet as ATM ARP Clients of this Server. On each client ATM interface, you will also need to pre-configure the ARP server's ATM address, so that it can set-up a direct ATM connection when making an ATM ARP_Request. As soon as they are on the network, these Clients will automatically register to the ATM ARP server to update its ARP cache table with their IP-to-ATM mapping information.

IP Router: if you intend to connect to other ATM or non-ATM IP subnets you will also need an IP router.

- Figure 12: The Classical IP over ATM Protocol Stack -

Which Benefits?

Performance: Besides allowing IP applications to run over the ATM network, the first major advantage of Classical IP is performance: 10/100Mbps/1Gbps Ethernet LANs limit the maximum packet transmit size (MTU) to 1500 Bytes. Classical IP provides 9180 Bytes by default and up to 64KB. This basically means that more data can be carried in fewer packets, thereby reducing the packet-processing by both the sending and receiving end systems CPUs, especially for bulk data transfers.

Simplicity: Due to its connection oriented nature, ATM removes the need for configuring multiple IP subnets (or LIS); this would typically cause congestion on traditional LANs due to non-contained broadcasts. Having one large ATM Logical IP subnet also simplifies the network's configuration, monitoring and management, without affecting performance.

Which trade-offs?

Only IP: As the name suggests, Classical IP only supports point-to-point applications based on IP.

Broadcasts: Classical IP does not support IP broadcasts. Since the source does not know who else is on the subnet, it cannot open point-to-multipoint connections corresponding to IP broadcast messages (as opposed to Ethernet, where the source does not need to know who else is on the subnet in order to broadcast). Therefore, Classical IP does not allow the use of several IP applications which require "broadcast" addressing (e.g. bootp) and operate over shared-media.

IP Router: interconnection of an ATM LIS subnet to the existing legacy LANs (e.g. Ethernet) or to another LIS is only provided via an IP router, performing software packet switching at the IP level, which is deemed to be the source of performance bottleneck within the

network, and defeats the purpose of ATM switching.

Single Point of Failure: Classical IP is not the optimal solution for critical network environments that require high availability: the ATM ARP server is a single point of failure for any LIS, since it maintains the mapping information required for any client-to-client SVC communication. Classical IP cannot manage dynamic IP to ATM address mapping reconfiguration easily.

Evolution of Classical IP

MARS (Multicast Address Resolution Server): The MARS protocol [13] addresses the **broadcast** limitation of Classical IP by providing "multicast" functionality based on two solutions:

VC Mesh: creating a mesh of point-to-multipoint VCs between members of a multicast cluster

MARS Server: using a Multicast Address Resolution Server. This MARS server has VCs to each member of the multicast cluster, and therefore is capable of performing the multicasting function for them. The boundaries of this Multicast cluster could extend across LIS boundaries.

Multicasting (i.e. opening simultaneous connections to one, two or many different destinations) is a very desirable feature for new applications such as Videoconferencing (any to any), Tele-teaching (one to any) or document distribution (one to any).

NHRP (Next Hop Resolution Protocol) [14] is aimed at removing the performance bottleneck linked to **IP routers**. Initially, an internetwork (e.g. inter-LIS) data connection will transit via the router as expected. At the same time, an NHRP request is sent along the same path to the Next Hop Server (NHS). This server can do IP-to-ATM address resolution across LIS boundaries. By supplying the remote host's ATM address, it allows the initiator to set up a "shortcut" (i.e. bypassing the router). The limitation with NHRP is that IP routers are still required when interconnecting ATM Classical IP subnets to legacy IP networks such as Ethernet.

- *Figure 13: NHRP Route Resolution* -

7. LAN Emulation

Why LAN Emulation?

In order to use ATM in today's environments and propose a smooth migration path for customers, ATM should be able to support all current networked applications. It should also integrate in today's network infrastructures efficiently.

Classical IP has partially addressed the problem by proposing a simple 'Network Layer' solution (OSI Layer 3) capable of supporting most IP applications over ATM, and interconnecting ATM to legacy networks through IP routers.

LAN Emulation (LANE), as defined by the ATM Forum [7], proposes a more comprehensive "DataLink Layer" solution (OSI Layer 2), aimed at providing the following objectives:

- a protocol capable of emulating legacy network mechanisms (including broadcasts), so that all current applications (e.g. IP , IPX, at present) that run over the Data Link Layer, would also run transparently over ATM nodes.
- a protocol 'glue', capable of bridging ATM to legacy networks at the Data Link Layer (OSI Layer 2, e.g. MAC), for efficient switching performance between both networks.

What is LAN Emulation?

With ATM LANE, the ATM cloud can be divided into multiple, separate Emulated LANs (**ELAN**s). Each of these would simply be identified by its name (e.g. 'fin' or 'eng') and operate as a self-contained broadcast domain.

The ability for ATM end systems to emulate LAN devices is provided by a group of servers on each ELAN, generically called **LAN Emulation Services**.

Each ATM node is considered a Lan Emulation Client (**LEC**) to these LAN Emulation services for a specific domain or ELAN. The breakdown into ELANs being arbitrary, each host can belong to one or more Emulated LANs (similar to 802.1 "Virtual" LANs).

Layer 2 bridging (or frame switching) with legacy Ethernet or Token Ring networks is achieved by "**Edge Devices**', which can, for instance, be an Ethernet switch with an ATM uplink. When crossing ELAN broadcast domains, Layer 3 **Routers** become as necessary as on typical LANs. *- Figure 14: LAN Emulation Network Topology -*

How does LAN Emulation work?

MAC over Ethernet: over Ethernet, hosts are identified according to their **MAC** address (OSI Layer 2). When any Network Layer (OSI Layer 3) application (e.g. IP, IPX, NetBios) wishes to reach a remote host, it will need to figure out the remote hosts' MAC address, either in its own hosts' cache table or by **broadcasting an ARP** (e.g. IP_ARP) request to all hosts on the subnet. The remote host will reply to the ARP with its MAC address and the connection will be made.

MAC over ATM: the purpose of LANE is to emulate this mechanism over ATM and end up with an ATM connection. In order to do so, several pieces of the puzzle are missing:

a- Broadcast: the Network Layer application (e.g. IP, IPX) wishing to reach a remote host will need to **broadcast an ARP** (e.g. IP_ARP) in order to figure out the remote hosts' MAC address. As previously indicated, ATM does not directly support Layer 3 (e.g. IP) broadcasts. Instead of having all hosts broadcast to each other, LANE has singled out one device which will handle broadcast requests from all other hosts on the subnet. Its role will simply be to broadcast the packets it receives (including ARP requests) to all hosts on the subnet: it is called the **Broadcast and Unknown Server (BUS)**.

At this point, the source can send (1) an ARP Request to the BUS, who will broadcast it (2) to all hosts on the subnet. The destination will recognize its own IP address and will reply to the source (3) via the BUS, giving it its own MAC address . But something else is missing to set up an ATM connection:

b- MAC-to-ATM mapping: in order to set-up the ATM connection, the source needs an ATM address, not a MAC address. It only knows the destination's MAC address so far but cannot map it to the ATM address. LANE has singled out another device on the subnet who will maintain the MAC-to-ATM mapping information for all hosts on the subnet; its role will be to handle MAC-to-ATM Address Resolution requests (also called LE_ARP requests) from clients on the subnet: it is called the **Lan Emulation Server (LES)**.

At this point, the source sends an LE_ARP request (4) to the LES, who will reply (5) with the destination's ATM address and the ATM SVC connection linking the two hosts will be established (6), over VPI/VCI (0, 197) for instance.

Here is a summary of the communication between two hosts or **LAN Emulation Clients (LEC)** on a subnet as described above:

- Figure 15: The LAN Emulation Connection set-up -

Although the connection has been set-up and hosts actually communicate, in order to *really* emulate a LAN, something is still missing:

c- Ease-of-Use: in order to benefit from the services of the BUS and LES, any host on the subnet must manually pre-configure parameters such as LES 40-digit ATM address, the ELAN name or its traffic characteristics (e.g. MTU size). LANE has singled out a third device whose role will be to automatically supply these parameters to all hosts: it is called the **LAN Emulation Configuration Server (LECS).**

Joining an Emulated LAN:

In order to join an ELAN and benefit from the LECS/LES/BUS servers, a host (LEC) will get in touch with the LECS and get all the information it needs for the specific ELAN it wishes to join.

Contacting the LECS (read: setting-up an ATM connection) has also been made easy, with 3 alternatives:

asking to use an ATM address called "**well known**", to which any LECS is expected to answer.

automatically downloading the LECS ATM address from the network using the **ILMI** protocol (as explained earlier).

entering the ATM address of the LECS **manually**.

Contacting the LES: having obtained the LES address from the LECS, the LEC will register its own MAC-to-ATM address mapping with the LES. It will also ask the LES for the BUS's ATM address.

Contacting the BUS: having obtained the ATM address of the BUS, the LEC will ask the BUS to include it in its broadcast distribution list (as a leaf of the BUS's point-to-multipoint connection).

The LEC has now fully become part of the ELAN. It can run any Layer 3 application and exchange data just like any Ethernet host.

What about Ethernet or Token Ring hosts?

We have stated at the beginning of this section that LANE was also designed to efficiently interconnect ATM with legacy LAN devices (Ethernet or Token Ring). This is accomplished by "**Edge Devices**". An Edge Device is basically a Layer 2 frame switch that does Cell to Frame conversion:

on the **ATM** side, it acts as a LEC and sets up ATM connections.
on the **Legacy LAN** side, it acts as a normal LAN port or segment switch. It forwards all broadcast Layer2 frames to the legacy devices or hosts attached to its LAN ports.

The Edge Device works as a **Proxy LEC** for all Legacy LAN hosts attached to it; meaning that it performs the functions of a regular LEC for the sake these non-ATM devices:

Joining the ELAN: the Proxy LEC will register itself to the LES as such, meaning that upon request, it will provide the LES with the MAC address of any of its Legacy-attached hosts.
Setting up ATM connections: when a Legacy LAN host wishes to contact an ATM device on that ELAN, the Proxy LEC will set-up the ATM connection with its own ATM address. When an ATM host wishes to contact one of the Edge Device's Legacy hosts, the Proxy LEC will provide the LES with its own ATM address, accept the connection, and forward the data to that legacy host.

What is needed to run LANE?

ELAN: you will need to segment your network into logical broadcast domains or Emulated LANs (ELAN). They would be interconnected at Layer 3 by Routers. An ELAN could comprise both ATM and Legacy-attached devices behind Edge Devices.

LECS: to make your whole network configuration easier, you will need at least one LAN Emulation Configuration Server (LECS). It will maintain a database with the characteristics of all the ELANs it manages, including the ELAN names, LES ATM address, MTU size, etc,...

LES: on each ELAN, you will need a LAN Emulation Server (LES). Its main job is MAC-to-ATM address resolution for unicast traffic (just as an ATMARP Server does IP-to-ATM mapping with Classical IP).

BUS: on each ELAN, you will also need a Broadcast and Unknown Server (BUS). Its job is to forward all multicast, broadcast and unknown unicast packets, including IP_ARP requests.

LEC: you will need LEC driver software on each of your system's ATM adapters. This software device driver will intercept network access operations of higher layer protocols and translate them into ATM LAN Emulation operations. Your host could have multiple LEC logical interfaces and belong to multiple ELANs (i.e. 802.1 VLANs).

Edge Device: it would be used to bridge ATM and legacy clients (Ethernet or Token Ring) on your ELAN.

Router: it would still be needed if you intend to connect to other ELANs or non-ATM subnets.

Collectively, the LECS, LES and BUS are known as the **LAN Emulation Services**. These services may be implemented in ATM switches, edge devices, routers, or even in ATM hosts. They are either part of the device ATM software driver, or partially implemented in hardware for better performance (e.g. the BUS). They could be collocated, i.e. bundled together within the same device.

- Figure 16: The LAN Emulation Client Protocol Stack -

Which Benefits?

Functionality: LANE allows ATM hosts to run most current networked application, on top of the OSI Data Link Layer (e.g. MAC Layer 2). This includes IP, IPX, APPN, NetBIOS, or Appletalk.

Performance: LANE provides high performance bridging (at OSI Layer 2) between ATM and legacy LANs, through the Edge Devices. This allows existing legacy LAN infrastructure to also benefit from the switched performance of ATM, since routers are not needed within ELANs.

Another benefit is that LANE allows you to use higher MTU sizes (up to 9KB), as long as it is not limited by legacy devices on the ELAN (e.g. MTU of 1.5KB with Ethernet hosts on the ELAN).

Ease-of-use: LANE has been designed to be as simple to use as legacy LANs (e.g. Ethernet). On their ATM Client (LEC) interfaces, end-users only need to configure the name of the ELAN they wish to join (e.g. 'fin' or 'eng') and optionally the network address (e.g. IP). The rest can be done by the LAN Emulation Services.

Which Trade-offs?

Single Point of Failure: The **LAN Emulation Services** are a Single Point of Failure on any Emulated LAN. Many vendors already offer proprietary solutions to circumvent this limitation of the LANE1.0 specification [7]:

- Redundant LANE Services allow the migration of the Services to a distinct physical device, transparently for clients on that ELAN and/or
- Redundant LEC interfaces can also allow a host to remain active by automatically switching to a functional backup ELAN.

These solutions have been facilitated by LANE broadcasting which allows the dynamic reconfiguration and easier management of redundant equipment.

Scalability: LANE provides the ability to bridge ELANs across ATM and non-ATM networks. It also allows ATM devices (e.g. a widely accessed file server) to belong to multiple ELANs and benefit from the performance of LANE across these ELANs. Nevertheless, the scalability of switched ATM enterprise backbones is limited by the fact that **routers** are still needed to interconnect Emulated LANs or other non-ATM subnets across the backbone.

Evolution of LANE

LANE2.0: LANE2.0 provides a standard specification for redundant LANE Services within an ELAN, in order to remove this **Single Point Of Failure**. It also defines the LE-NNI protocol which coordinates transactions and load-sharing between the multiple LECS/LES/BUS servers within an ELAN. LANE2.0 will also provide better support of ATM Quality of Service, including ABR.

MPOA: in order to provide better **scalability** across switched ATM backbones, LANE can rely on MPOA to establish shortcuts and avoid routers across subnets, as it will be seen in the next section.

8. MPOA

Why MPOA?

LAN Emulation addresses the legacy interconnect issues of ATM by allowing ATM hosts to efficiently connect to non-ATM hosts and run non-ATM applications. But like Classical IP, it makes flat switched ATM networks similar to traditional LANs by segmenting them into internetwork subnets (such as IP subnets or VLANs) separated by routers. Inter-subnet routers introduce latency and prevent user data from benefiting from ATM switching performance achieved by LANE at the level of each subnet (e.g. within a VLAN).

The goal of **Multiprotocol Over ATM** (MPOA) [8] is to address these scalability limitations and maintain the high-performance benefits of switching technologies such as ATM across larger networks, and across subnet boundaries. It complements LANE for internetwork communications, as within a subnet, peers would continue to communicate using LANE. It optimizes the use of routers by interrogating them only once to find the route between two hosts, and then letting them communicate over the high speed switched route.

How does MPOA work?

When you drive to someone's house, you will only spend time asking for directions the first time you go there. Likewise, MPOA will only send the first packet through the router in order to 'discover' the fastest switched path, after that, all subsequent packets will be able to avoid the router and use the 'discovered' high speed route to the destination, even if it is in a separate subnet. In short, MPOA removes the need for systematic route discovery and processing when sending packets across network boundaries. Why is it called MPOA? Because this protocol enables hosts using different network protocols (e.g. IP) to communicate across different networks (including IEEE802.1 VLANs), without requiring a router to carry the information, and translate what is said.

When a host wants to set up a connection with another host on a different subnet, it will invoke the assistance of a an MPOA server on its subnet, supplying it with the Network (e.g. IP) address of the remote host. To set-up the ATM connection, the MPOA server will need to find the ATM address which corresponds to the network address of the remote host belonging to another subnet. Using NHRP, it will forward this mapping request to MPOA servers on other subnets until one replies with the network address-to-ATM mapping. At this point, it will be able to set-up an ATM connection between the two hosts,

using the high speed switched route it found when it reached the remote host's MPOA server.

Here is a summary of how MPOA works:

Configuration: The source (MPC) contacts the LECS , as it would with LAN Emulation, in order to obtain its configuration parameters (e.g. period before sending a data flow over the shortcut).

Local Discovery: The source (MPC) discovers the ATM address of the MPOA server (MPS) on its subnet and vice-versa by using LAN Emulation ARP requests (LE_ARP).

Local Connection: The source (MPC) wishing to connect to another client on the same subnet would use the LAN Emulation mechanisms as described in the previous section.

Distant Connection: If the destination is not on the same subnet, the source will start the MPOA procedure: It will contact its Ingress MPOA Server (1) in order to ask for the shortcut to the destination.

Target and Route Resolution: The Ingress MPOA Server (MPS) will issue an NHRP request (2) to find the MPOA Server belonging to the destination's subnet. For route resolution, MPOA can rely on traditional route discovery protocols such as OSPF.

Once discovered, the destination's or Egress MPOA Server will be able to provide the source's ATM address to the destination MPC, using a Cache Imposition Request (3). The destination MPC will optionally provide its MPS with a tag to identify the shortcut (4), which will relayed it back to the source's MPC with the NHRP reply. At this point, the shortcut can be established (5).

Data Transfer: According its type, data can be transferred as a shortcut flow using the shortcut established with MPOA, or as a default flow using the long routed path for non-flow oriented traffic such as SNMP.

- Figure 17: MPOA Default and Shortcut Data Paths -

What is needed to run MPOA?

MPS: You will need an **MPOA Server** (MPS) in each of your ATM subnets. Typically, this MPOA Server functionality is implemented in software and/or hardware in a device on the subnet (e.g. an ATM switch, edge switch or router). As part of the MPOA Server functionality, you may also need an NHRP Next Hop Server **(NHS)** that will be capable of answering the MPOA server's NHRP inter-subnet resolution requests, as well as a **Router**, which could help in route calculation, using standard Internetwork Layer routing protocols.

MPC: If you have ATM end-systems (e.g. servers), they will need to behave like **MPOA Clients** (MPC). Typically, this will a be software functionality provided in your ATM NIC driver which will allow you to communicate with the MPOA server.

If you have non-ATM hosts (e.g. Ethernet PCs), they will not require any change in their configuration, but will need someone to be a **proxy MPOA Client** for them. Typically, the edge switch which connects their network to ATM will proxy for them.

MPOA uses the following protocols between MPOA Servers and Clients:

UNI Signalling (UNI 3.0, 3.1, or 4.0).
LAN Emulation (LANE2.0).
Next Hop Resolution Protocol (NHRP).

What are the benefits?

Scalabiltity: This is the main benefit MPOA Internetwork switching provides to ATM. It allows ATM backbones to span across multiple subnets and effectively using ATM switching across subnet boundaries.

Legacy LAN switched interconnect: It also allows ATM to interconnect legacy subnets such as Ethernet VLANs or Token Ring LANs over a fast switched infrastructure, without requiring router for data forwarding. Although Classical IP and MARS/NHRP would also allow switching between ATM subnets, a router would still be needed when leaving the ATM infrastructure.

Virtual Routing: It rationalizes network designs by attributing clear tasks to network devices most adapted to handle them: route calculation is done in intelligent routers and fast forwarding by the ATM and edge switches.

What are the trade-offs?

Complexity: MPOA brings complexity to simple switched networks, just to allow them to span across subnet boundaries. MPOA makes sense only over large networks with multiple subnets.

Performance: MPOA route calculation at connection set-up may introduce latency. Although it would improve performance for flow-oriented traffic (e.g. a large file transfer where only the first packet would be routed), it will not benefit non-flow oriented communications such as SNMP which will continue to use the default routed path.

Unicast: MPOA only addresses unicast data; no provision has been made for multicast (or broadcast) so far.

MPOA is a complementary protocol to Classical IP and LANE. It should be only be used where most appropriate: for flows crossing subnets boundaries.

9. Conclusion

ATM emerged from a major standardization effort, following work from Telecom laboratories on both sides of the Atlantic: all had realized the values of cell switching for wide-area multimedia networks. Although ATM was chosen as the basic building block for next generation telephone networks (B-ISDN), lately, the most enthusiasm in this new technology has come from data network applications. Industry consortia such as the IETF and the ATM Forum have made the standards evolve in order to better reflect today's most urgent needs for ATM: its use for transporting current data applications such as IP and for interconnecting legacy networks such as Ethernet.

Although wide-area ATM networks are just starting to appear and native ATM applications currently have limited scope, ATM is being used in data networks, today. Classical IP has made IP applications available to ATM hosts. LAN Emulation has provided an efficient way of interconnecting ATM to legacy networks and run legacy applications end-to-end. MPOA has enabled switched ATM backbones to provide full

value despite being segmented into multiple subnets.

It is unlikely that ATM will be the panacea it was once thought to be, and replace the installed base of data, voice and video devices and services, over Local and Wide Area Networks. It is just as unlikely that ATM it will not be implemented where it provides the most value in today's network environments, and help them evolve towards next generation integrated infrastructures.

Appendix A. ATM Standards

The ATM Forum (consortium of vendors, users and service providers), IETF and ITU-T have specified ATM standards and are still making them evolve.

ITU-T

The ITU-T has provided the following recommendations:

[1] I Series: including ATM Layer (I.361), Adaptation Layer (I.362, I.363) Specifications, and ATM-layer Transfer Capability (I.371)

[2] Q Series: including (Q.93b, Q.2931, Q.2110): ATM signalling definition

ATM Forum

The ATM Forum has provided the following standards:

[3] UNI 3.0, 3.1, 4.0: based on the ITU-T Q Series, UNI standards are subsets of Q.2931 plus additional specifications for point-to-multipoint.

[4] P-NNI1.0: Private Network-Network Interface and Signalling Specification.

[5] ILMI 3.0, 3.1, 4.0: Integrated Local Management Interface

[6] Traffic Management Specification version 4.0

[7] LAN Emulation over ATM version1.0 (work on 2.0)

[8] MPOA Baseline version 1: MultiProtocol Over ATM

[9] Native ATM Services: Semantic Description version1.0 / ATM Programmatic Interface

IETF

The IETF (Internet Engineer Task Force) has published most specifications for IP over ATM:

[10] Classical IP and ARP over ATM (RFC 1577)

[11] ATM Signalling Support for IP over ATM (RFC 1755)

[12] Multiprotocol Encapsulation over ATM Adaptation Layer 5 (RFC 1483)

[13] MARS: Support for Multicast over UNI3.0/3.1 based ATM Networks

[14] NHRP: Next Hop Resolution Protocol

[15] ATM Network Management: MIB (Management Information Base) for Network Management of TCP/IP based internets (MIB-II)