7

Integration of IP and ATM

Before ATM, most data networks, such as Ethernet and token ring or FDDI networks, provided connectionless data services. They transmitted data using common MAC, LLC, or bridging protocols without setting up connections before sending data. Connectionless network technology such as TCP/IP also were connectionless service networks using the datagram routers. Connectionless networks are advantageous in that the end systems place no burden on connection management, thereby simplifying the design. Also, as discussed in Chapters 1 and 3, there are various other advantages in using connectionless networks.

In contrast, ATM networks are connection-oriented networks. In ATM networks virtual circuits are established between two end systems, and cells are switched according to their connection identifiers. Since resources are statistically allocated per connection basis, ATM networks can provide guaranteed QoS for connections. In addition, since routing is done only at the connection setup phase, ATM networks can support cell-sequence integrity.

Even though ATM networks were designed for connection-oriented services by nature, they are required to provide connectionless services as well, so that they can interoperate with other existing connectionless networks and services. Today many ISPs today use ATM network cores to transport TCP/IP traffic.

There are two major reasons for the importance of transporting IP traffic over ATM networks. Initially there was a cost-performance advantage as

ATM switches were widely available and cheap. While routers were also widely available, the link types traditionally supported by routers were either low in speed (e.g., T1 or T3) or not suitable for long-distance transmission (e.g., Ethernet). This first advantage has now disappeared in many cases due to the spread of cheap multifunctional routers with high-speed I/O ports. Second, ATM networks have the built-in ability to do bandwidth management and traffic engineering, mostly due to their connection-oriented attribute. This enables automatic distribution of traffic in contrast to the case of datagram routing, which does not utilize multiple parallel paths well. This second advantage is still valid as a key advantage that ATM technology has over IP technology. It can be viewed as derived from the fundamental differences in connection-oriented versus connectionless technologies.

There are also some other merits to using IP over ATM. While ATM supports high-speed links and multiplexing, it also supports multiplexing low-speed links into high-speed ATM links, thereby affording an efficient multiplexing solution for many edge-switching systems. As ATM supports various network management functions such as fast automatic rerouting and fault recovery, network management becomes easier. In addition, efficient traffic engineering and traffic control may be achieved by using various mechanisms defined in ATM technology.

This chapter discusses the integration of IP and ATM. We first describe various integrated/overlay architectures such as *classical IP over ATM* (IPoA), MPoA and MPLS over ATM, and then discuss the routing, multiplexing and switching, network control, traffic management, and QoS issues related to them.[1]

7.1 Concepts and Architecture

The problem of supporting TCP/IP protocols over ATM networks has been extensively studied during the last decade. The resulting architectures can be categorized into the *overlay model* and the *integrated model* depending how deeply their network control, signaling, and routing planes are entwined.

1. It should be noted that while both MPOA and MPLS contain the term "multiprotocol," they were both essentially designed for the support of IP. In the case of MPOA, support for other network layer protocols such as IPX was important initially but the interest has decreased considerably. In the case of the MPLS, the opposite has happened: While originally conceived as a way of improving IP technology, it is now being recognized as a technology that may be applied in a more generic manner, with optical networks being an obvious example (refer to Section 8.4).

In the overlay model it is assumed that the IP and ATM network nodes are unaware of each other. The ATM network appears as a cloud to the IP network, as is shown in Figure 7.1(a). The IP traffic is transported over ATM pipes between IP routers. The IP routers are unaware of the ATM switches that interconnect them. They only know of the ATM pipes (or virtual connections) that connect themselves to other IP routers. Basically, the routing protocols that run in the IP and the ATM networks are independent. The IP level routing protocols indicate how to go from one IP node to another, while the ATM routing protocols supply the same information for the ATM nodes. For example, ATM routing protocols may be used to find the correct route for the ATM virtual connection from an IP router *A* to another IP router *B*, but from the IP router's perspectives, routers *A* and *B* are only a single link apart. The two routers are unaware of the numerous ATM switches used to relay the virtual connection between the two routers.

The use of independent routing protocols also implies that there is no relation between the addresses from the ATM and the IP networks in the overlay model. There is also no algorithmic method for translating between the two types of addresses. From the IP network's perspective these facts mean that an ATM network will look like a network "cloud" offering connectivity to other points on the edges of the ATM network leading to other IP routers. However, the IP routers do not know how those connections are made through the ATM network. Such a structure implies the need for a protocol that is able to map the destination ATM address to the destination IP address. This is a basic problem that all the overlay architectures must solve. Later in this section we will examine various overlay architectures that differ primarily in how these problems are solved.

In the case of the integrated model, the IP and ATM nodes are peers; when ATM and IP networks interoperate, the ATM nodes and IP nodes are

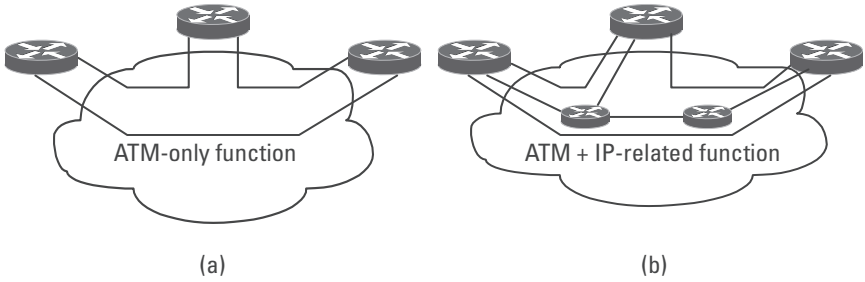

Figure 7.1 TCP/IP over ATM service models: (a) overlay model, and (b) integrated model.

specifically aware of each other. When IP traffic is transported over an ATM network, the IP router at the edge of the ATM network is fully aware of the ATM network's structure. In general, the IP-level routing protocols indicate how to go from one IP node to another, while the ATM routing protocols supply the same information to the ATM nodes. However, as shown in Figure 7.1(b), the IP router on the edge of the ATM network is aware of the structure of the ATM network. Consequently, it can route the IP packet to the next ATM switch along the path, instead of blindly putting it into an ATM virtual channel. In addition to the routing protocols, the signaling protocols used in the two networks must also interoperate. This may require new signaling schemes, such as the *label distribution protocol* (LDP) developed for use in MPLS-capable ATM networks. The integrated model implies that the ATM addresses and IP addresses can be translated from one to another by algorithmic methods, which is an obvious consequence of the interoperability of the routing protocols used in each network. As ATM addresses and IP addresses are translatable, additional address resolution protocols are not necessary to find out the destination ATM address when only the destination IP address is known. Examples of the integrated model include the integrated PNNI model from the ATM Forum and the MPLS protocol suite developed by the IETF.[2]

Figure 7.2 shows various methods for integrating IP networks and ATM networks. Basically, LANE, classical IPoA, MPOA, and NHRP methods belong to the overlay model, whereas the *integrated PNNI* (I-PNNI) and MPLS methods belong to the integrated model.

7.1.1 Classical IPoA

The basic classical IPoA method was defined by the IETF for transporting IP traffic over ATM networks. The basic method specifies how to operate an ATM network as a single IP subnet, or specifically, how to connect two nodes on the same IP subnet directly, by an ATM connection. However, connecting two nodes on different subnets requires routers. As such, the IPoA specified in RFC1577 does not change the fundamental nature of the IP protocol, so it relies on IP routers for interconnecting subnets consisting of LANs.[3]

2. The development of the integrated PNNI method, however, has not been done actively enough, so discussions will mainly be concentrated on the MPLS method in later sections.

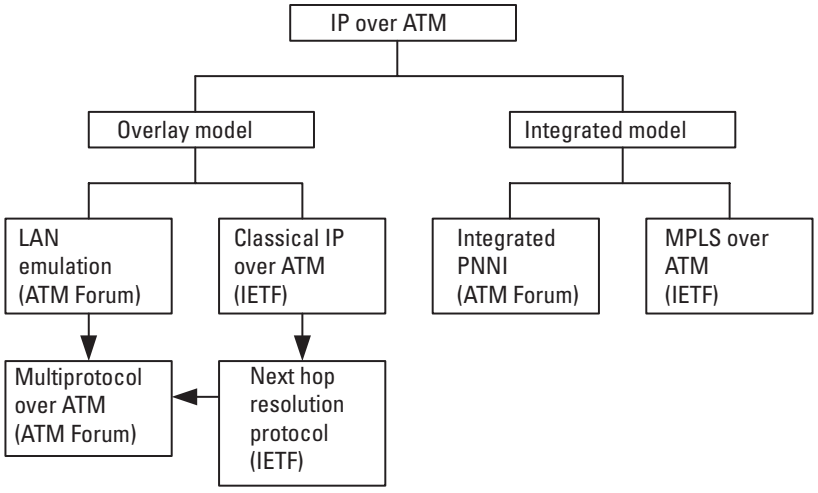

Figure 7.2 Taxonomy of IPoA models.

7.1.1.1 Elements of Classical IPoA

The classical IPoA approach is based on a special type of IP subnetwork called a *logical IP subnetwork* (LIS). An LIS corresponds to the concept of a subnetwork in traditional Ethernet LANs except that it consists of hosts and routers that are connected through an ATM network. An LIS consists of hosts and routers that have the same subnetwork mask and the same subnetwork address. Any two hosts in the same LIS communicate directly, but hosts in different LISs can communicate only through a router even if a direct ATM connection can be established between them.

As the LIS is an IP network constructed over the ATM network, an address resolution functionality is needed. In classical IPoA the *ATM address resolution protocol* (ATMARP) server provides this function. The ATMARP server basically allows a client to find the ATM addresses corresponding to a given IP address. Note that this ATMARP function only maps between ATM addresses and IP addresses. No other type of network level protocol,

3. The classical IPoA methods are defined largely in two main RFCs, RFC1577 and RFC1626. Recently these two RFCs have been updated and unified into a single specification RFC2455. Additionally, RFC2332 (formerly RFC1735 but currently upgraded to support UNI 4.0 features) defines how UNI 4.0 signaling methods and parameters may be used to set up and release SVCs for transporting classical IPoA IP traffic.

such as IPX or AppleTalk, is supported. This exhibits the IP-centric nature of the classical IPoA solution.

It is possible to construct multiple IP subnets over a single ATM network. Each of these subnets will constitute an LIS and, for communications between nodes on the same subnet, only the ATMARP function is required. However, for communications between hosts on different subnets, routers are needed to connect with other subnets. The current classical IPoA specifications also support IP multicasts through the *multicast address resolution server* (MARS) standards defined in RFC2012.

Classical IPoA operation over PVCs. The simplest mode of operation of a classical IPoA network is to use *permanent virtual channels* (PVCs) among all the routers connected by the ATM network [1]. This means that each router will have a PVC connection to all its neighbors. When the routers are powered on, all of them use the inverse ARP [2] protocol to find the IP address of the host on the other end of each of its default PVCs. Once all the IP addresses become available, the routers can use various routing protocols, such as RIP or OSPF, to find out routes to all the other points in the overall network. There are two major drawbacks to this PVC approach: It relies on the router being manually preconfigured to set up the PVCs, and it is not scalable to a large size. Manual configuration is not practical when the network grows to a large size.

Classical IPoA operation over SVCs. The use of *switched virtual channels* (SVCs) requires the setup of SVCs on demand [1]. To establish a VCC between two hosts in an LIS, a mapping is necessary between the ATM addresses and IP addresses of the source and the destination. Based on this mapping, IP addresses are resolved to ATM addresses through the ATMARP and the inverse process is done through the *inverse ATMARP* (InATMARP). For such address resolution, each host must register its IP address and ATM address to the ATMARP server in the same LIS. Then the ATMARP server can resolve the IP addresses in the same LIS.

Figure 7.3(a) shows an example of the classical IPoA model using an ATMARP server and Figure 7.3(b) shows the protocol stack of the classical IPoA. In Figure 7.3(a), Host 1 (IP address A, ATM address X) wants to send some connectionless data packets to Host 2 (IP address B, ATM address Y). So Host 1 sends an ARP request to the ATMARP Server 1 in the same LIS. Then ATMARP Server 1 resolves IP address B to ATM address Y and then sends an ARP response to Host 1. Next, Host 1 sets up a VCC to Host 2 using the ATM address Y. Packets are transmitted over this virtual connection

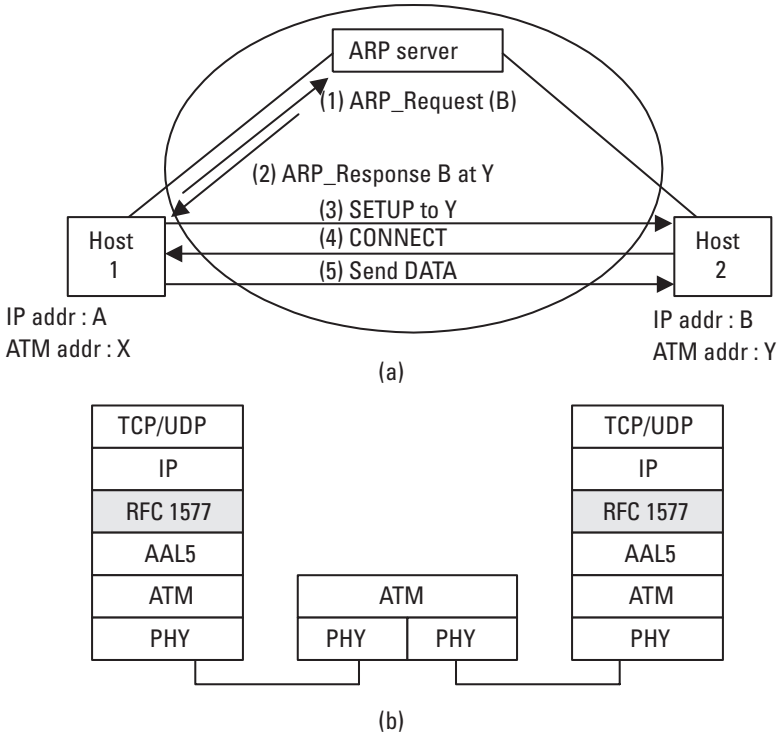

Figure 7.3 Classical IPoA: (a) operation example, and (b) protocol stack.

to reach Host 2. Host 1 preserves the mapping information indicating that IP address B is mapped to the ATM address Y for use in the next packet transmission. Because all these procedures are done below the actual IP layer, the IP layer need not care about the specifics of the ATMARP server interactions or the ATM connection setup procedures. Consequently from the IP layer point of view, there is no difference from transmitting data over traditional LAN or other data link protocols.

For this design to be robust, a number of timers are needed. First, the ATMARP server uses a timer to periodically test that a previously registered host is still alive. Whenever the timer goes off, the ATMARP server uses an InATMARP message to check that the host is still responding. If the host fails to respond the SVC will be torn down. Second, another timer is an inactivity timer used to test if data flow over an SVC setup between two hosts discontinues for a certain amount of time. If this happens, the connection is

automatically released. Essentially, the above two timers are to release the unused VCs in a timely manner such that no VCs are wasted.

Note that the connection setup between Host 1 and Host 2 is an SVC. This is the key difference from the PVC method. Compared with the use of PVCs, the use of SVCs is more efficient as it would set up and use a virtual channel only when needed. This saves VPI/VCI label space and enables network resources to be utilized when needed. Saving label space is important in larger networks with many connections and in the networks where the hardware of practical ATM switches limits the number of available VCs. The use of timers is an important way of ensuring that these resources are not wasted.

Classical IPoA between different LIS networks. In classical IPoA networks, traditional hop-by-hop IP routing is used when routing between different LIS networks. Even when it is possible to use a direct ATM connection to connect two communicating hosts, if the two hosts are on different LISs, the packets must be transmitted through router(s) that connects the two LISs. This is a key concept in the classical IPoA model, which contributes to the term "classical" in the name. Accordingly, if two hosts are internal to an LIS (the *intra-LIS* case), it is possible to use direct ATM connections to communicate between two hosts. When those hosts lie on different LISs (the *inter-LIS* case), however, a hop-by-hop routing approach using routers is needed. Figure 7.4 demonstrates the classical IPoA operation when data flows

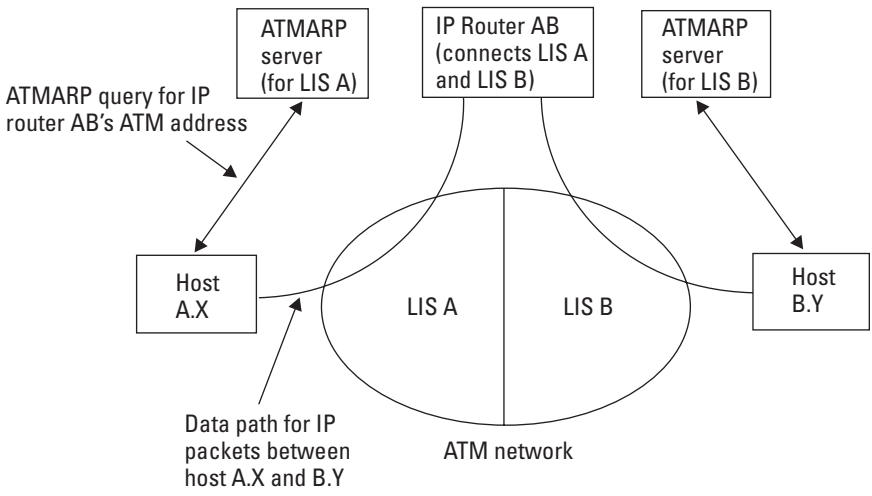

Figure 7.4 Operation of classical IPoA when data flows between different LIS networks.

between different LISs. For a host A.X on LIS A to send a packet to host B.Y on LIS B, the packet must be sent to router AB over an ATM connection between host A.X and the router AB. The router will then forward the packet to host B.Y again by using another ATM connection from itself to the host.

This method is inefficient because hop-by-hop routing over IP routers is needed for inter-LIS communication, even when a direct end-to-end shortcut is possible. For example in the network shown in Figure 7.4, the host A.X and host B.Y are on different LISs, but actually on the same ATM network. Thus, it is possible to set up a simple direct path between the two hosts instead of rerouting the packets every time they come. This inefficiency has triggered the development of the NHRP model, defined in the Section 7.1.2.

7.1.1.2 Classical IPoA and Encapsulation

Any method for transporting IPoA needs to address how to encapsulate IP frames in ATM cells. This is a basic problem for all IPoA models, and the basic solutions are outlined in RFC1483. According to this document, hosts can deploy two different methods to encapsulate different network layer protocols. The first method is to multiplex multiple protocols in a single ATM VCC. In this case the protocol of the carried PDU is identified by the LLC header and the *subnetwork attaching point* (SNAP) header. The other method is to set up a VC for each protocol. The first method is suitable for networks that use only PVCs, and the second method is suitable for the networks employing SVCs in which VCCs can be created or removed flexibly. These methods are further discussed in Section 7.2

The MTU must also be defined along with the encapsulation method. The actual MTU value may be negotiated while the connection is being set up by signaling. The size of the MTU that a host can transmit over the ATM virtual channel is 9,180 octets [3]. If all members in an LIS consent, this value can be changed. This is aligned with the default MTU defined for the *switched megabit data service* (SMDS) standards in RFC1209. Also all routers should use the IP path MTU discovery mechanism to find out the maximum MTU for a path [4].

Note that the address resolution–related messages, ATMARP request/response and InATMARP request/response, which were defined for classical IPoA operation, also need to have their packet formats and encapsulation methods defined. As these messages are not layer 3 packets but ATM-level messages, they are directly transported over ATM connections in LLC/SNAP-encapsulated format. The details and specific formats are defined in RFC1577.

7.1.1.3 Classical IPoA and ATM Signaling

Classical IPoA is based on the overlay model for IPoA transmission. As mentioned above, the IP routers on the edge of the ATM network view the ATM network as a cloud that offers the ability to connect with other IP routers connected to the ATM network. While the IP routers need not be aware of the ATM networks structure, the IP routers must be able to use the ATM signaling to set up connections to the destination IP routers. This means that the IP routers must be able to map IP parameters with the relevant ATM signaling parameters. RFC1755 specifies how UNI 3.0/3.1 signaling is used to support classical IPoA by defining how these parameters are mapped. (Refer to Section 7.4.2 for further discussions on signaling.)

7.1.2 NHRP

Although the classical IPoA method has the advantages of being conceptually simple and not requiring any change to existing systems, its performance is rather limited as communication among different subnetworks must be done through routers. This can cause a serious degradation of performance in an ATM network consisting of a large number of LISs. In ATM networks, hosts can communicate directly with each other without the involvement of IP layer switching in routers, and this fact can be exploited to enhance performance by removing unnecessary relay nodes.

As a means to set up direct connections in *nonbroadcasting multiaccess* (NBMA) networks such as ATM, the IETF has introduced NHRP, which relies on a new type of ARP server for ATM networks.[4] The aim of NHRP is to enable a source host to bypass all or some intermediate routers so as to establish a direct ATM connection to the destination host. When such a direct connection is set up, it is said that a "direct shortcut SVC" is used. In

4. The term NBMA comes from the fact that ATM networks allow multiple access but, unlike traditional LANs such as Ethernet, ATM networks do not allow broadcasting in a native manner. As pointed out earlier, this is one of the main differences between ATM networks and traditional LANs. While in our discussions we concentrate on the application of NHRP to IP and ATM networks, it must be noted that NHRP was defined not only for IP networks but also to support other protocols such as IPX and AppleTalk. As such, the NHRP specifications use the term internetworking address when referring to the address of the upper network layer's addressing scheme and use the term NBMA address to refer to the native addresses used in the NBMA itself. For the cases that we will consider, IP addresses correspond to internetworking addresses, while ATM addresses correspond to native NBMA addresses.

other words, NHRP is an interLIS address resolution protocol, a more complicated ARP that can be used in NBMA networks having multiple LISs.

Specifically, NHRP is used to determine the IP layer address and NBMA network addresses of the *NBMA next hop* toward a destination station. If the destination is connected to the NBMA network then the NBMA next hop is the destination host itself. Otherwise, the NBMA next hop is the egress router from the NBMA network that is "nearest" to the destination host. Usually this egress router would be the last router connected to the NBMA network that is on the routed path to the destination host.

7.1.2.1 Elements of NHRP

NHRP uses the concept of LIS introduced in classical IPoA. It is applied when a single NBMA network is divided into a number of disjoint LISs. In each LIS of the NBMA network, there is at least one *next hop server* (NHS) that resolves IP addresses to NBMA addresses. The NHS constructs an NHS address mapping table by utilizing information it gets through NHRP registration packets from clients on the same NBMA network or by applying dynamic address learning mechanisms. NBMAs as well as LISs are normally connected by routers, with the NHS usually coresiding in the inter-LIS routers.

NBMA stations are those stations that implement the NHRP protocol. They use the NHRP protocol to find the interworking layer address and the NBMA address of the NBMA next hop on the path to the destination host. NHRP stations can be divided into clients and servers depending on their operation, namely *next hop clients* (NHCs) and NHSs. All NHSs and NHCs maintain next hop resolution tables that map internetworking addresses to NBMA addresses.

Each NHS implements the NHRP protocol. Conceptually NHS may be considered as residing in a router. Normal IP routers can be both NHRP clients and servers. The NHRP specifications do not make any assumption that all routers implement the NHRP protocols. NHCs maintain an address cache that maps internetworking addresses to NBMA addresses. Examples of NHC are IP end hosts. For NHRP to operate correctly, all NHCs must know the address of at least one NHS. The NBMA address of this NHS may be obtained by various methods such as manual configuration or by using anycast addresses.[5]

Each NHC must register its NBMA address and IP address with its *serving NHS*. In the case of ATM networks this means that the ATM and IP address of the client is known by the serving NHS. The *last hop NHS* refers

to the last NHS along the routed path to a client. This NHS is usually the serving NHS of the destination host or egress router.

7.1.2.2 Operation of NHRP

The NHRP protocol relies on various messages (see Table 7.1). Among them the main types are the NHRP resolution request/reply and registration messages. Figure 7.5(a) shows the basic message flows for client address registration. Each NHRP station registers its NBMA address and internetworking address with the NHS by using an *NHRP registration request*. This message contains the NHC's ATM address, the NHC's IP address, and the NHS's IP address. When the NHS receives this message, it may start to construct a cache based on this information. The NHRP clients use *NHRP resolution request* and reply messages to find the NBMA address of the next hop server

Table 7.1
NHRP Messages

Message Type	Direction	Description
Next-hop resolution request	Station → NHS	Sent to NHS to find the ATM address of the destination
Next-hop resolution reply	NHS → Station	Reply to the next-hop resolution request
Registration request	Station → NHS	Registration of next-hop information
Registration reply	NHS → Station	Reply to the registration request
Purge request	Station → Station	Requests a removal of next-hop information from the cache
Purge reply	Station → Station	Replies to the purge request
Error notification	Station → Station	Notifies sender of error indications and related problem descriptions

5. An anycast address is a special type of address that can be used to find the nearest host or server that is listening to that address. A client may use this address to automatically find the servers when first booting up. Anycast addresses have been defined for ATM networks and IPv6 networks, but not in IPv4 networks.

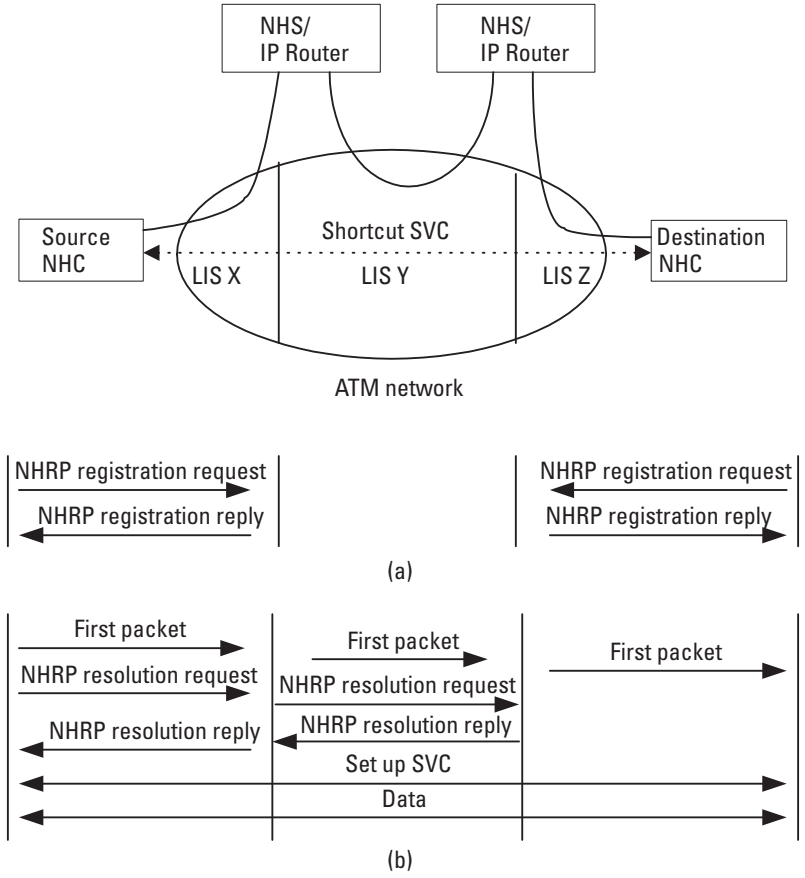

Figure 7.5 Operation of NHRP: (a) registration flow, and (b) address resolution flow.

on the routed path to the destination. The resolution request message contains [source NHC's ATM address, source NHC's IP address, destination's IP address].

Figure 7.5(b) shows the basic message flows for address resolution in NHRP. The basic procedure is as follows: When the source NHC has data to transmit it first checks its address cache to see if the destination host address has already been resolved. If not, the source host sends an NHRP resolution request to its NHS server. If the source host and the destination host are connected to the same subnetwork, the NHS will reply with the NBMA address of the destination host. If the destination host is not connected to the same

subnetwork, the NHS looks up the next hop router's address of the destination host in its forwarding table and forwards the NHRP request to the next NHS.

The same procedure is carried out by the next-hop NHS. If the NHS contains a mapping for the destination IP address in its cache, it returns an NHRP registration reply with the destination's IP address and NBMA address. As mentioned above, this final NHS router that generates the initial NHRP registration reply is called the *last-hop NHS*. The NHRP registration reply is usually returned along the same path that it took in reaching this last hop NHS, thereby allowing all the NHS to update their cache entries with the information regarding the destination.

Later on, other hosts may request address resolution for the same destination host. The new requests will be forwarded through the NHS to the serving NHS for the destination host. Before arriving at the last hop NHS the request may first arrive at one of the NHSs that had previously cached the reply sent back to a previous request for the same destination host. In such a case the NHS may send back a reply based on the data in the cache but with the data marked as *nonauthorative*. By having the NHSs reply based on the cached data, the performance of the protocol may be improved and the scalability may be also increased for the cases where the destination host is a popular destination. It is important to mark the data as being nonauthorative as the requesting host should be able to choose between using cached data and always getting data from the last-hop NHS.

The NHRP request is always forwarded along the normal routed path to the destination host according to the network layer routing tables. It is possible that the NHRP request may finally reach the egress router of the NBMA network without reaching the final destination host. In such a case the NHRP registration reply will contain the address of this final egress router or the next hop router. An NHRP request never crosses the border of an NBMA network.

If neither the destination address nor the next-hop router to the destination is found within the NBMA network, the last NHS to receive the request will send back a negative NHRP resolution reply with a code indicating that no entry was found.

7.1.2.3 Encapsulation and Interaction with ATM Signaling

As stated above, NHRP can be viewed as an advanced version of the basic classical IPoA model. As we have shown above, most of the changes were within the address resolution procedures and architecture. In contrast, the actual methods for the encapsulating IP packets over ATM connections

follow the basic classical IPoA methods. Also the basic interactions with the ATM signaling procedures are the same as the classical IPoA case. These are further discussed in Section 7.4.

As in the case of classical IPoA the NHRP specific messages are transported directly over ATM connections using AAL-5 framing with LLC/SNAP encapsulation. The details of the packet formats are described in [5].

7.1.2.4 Limitations of NHRP

As many researchers have pointed out, NHRP can suffer from poor scalability. At the NHRP client level, due to processing and memory limitations the NIC on the client may not be able to maintain the large number of mappings needed to support a direct ATM connection to each different destination. This problem would manifest itself not in the normal end user clients but in the nodes with a large amount of connections such as Web servers using an ATM interface.[6] At the NHRP server level, ATM-to-IP address mapping within large LISs would mean that the NHRP server or NHS would have to maintain a very large table of mappings. At the NHRP domain level, this leads to the connection scaling of the order of N^2 for the number of hosts on the network, N.

Another limitation of NHRP is that all the routers within an ATM network should be NHRP-aware. While the protocol does not require this, NHRP cannot resolve routing loops that can occur when NBMA networks and normal IP routers that do not understand NHRP are mixed

A third limitation is that when LISs have multiple NHSs, they need a mechanism to synchronize the cached information. While such a protocol [the *server synchronization cache protocol* (SSCP)] has been defined, any such mechanism is prone to failure and much harder to make robust. This may be due to the problems in the protocol design, but just as likely is from human errors in implementation or configuration.

A fourth limitation is that NHRP cannot set up multiple shortcut paths in ATM networks. This is because NHRP basically follows the destination-based routing paradigm used in traditional IP routing. It's therefore unable to set up multiple paths for different QoS and user requirements. This is especially disadvantageous when considering that ATM is capable of supporting this feature.

6. Note that though the Web server is a server with respect to its client, it is a client with respect to the NHRP protocol.

7.1.3 LANE

LANE was developed by the ATM Forum as a method of transporting LAN traffic over ATM networks. As the name suggests, the main idea was to emulate the behavior of popular LANs at the MAC layer, so that user applications could be run with minimal changes. This meant that LANE was designed to support the connectionless services offered by traditional LANs. Additionally, it supports broadcast and multicast services that all LANs offer. LANE currently emulates two LAN technologies, Ethernet (IEEE 802.3) and token ring (IEEE 802.5).

The two main types of LAN systems, Ethernet and token ring networks, have a number of common representative characteristics. First, messages may be characterized as connectionless, as opposed to the connection-oriented approach of ATM. Second, broadcast and multicast are easily accomplished through the shared medium of LANs. Third, LAN MAC addresses, which are basically the manufacturing serial numbers independent of the network topology, are a globally unique ID for whatever device that is used to connect to the LAN.

When the ATM Forum defined LANE across ATM networks the aim was to define an architecture and protocol suite that could offer the services based on the characteristics above. LANE was defined so that it could be implemented as a software layer in end systems, without affecting the layers above the MAC layer. Additionally, LANE supported the interconnection of ATM networks with traditional LANs by means of bridging methods. Consequently, LANE allows the interoperability between software applications residing in ATM-attached end systems and in traditional LAN end systems. In other words, LANE provides a simple and easy means for running existing LAN applications in the ATM environment. By offering different types of emulation at the MAC layer, LANE offers support for the maximum number of existing applications.

7.1.3.1 Characteristics of LANE

A significant characteristic of LANE is that it can support all network layer protocols. This is due to the fact that it operates below the MAC layer. In addition, LANE networks may be bridged with real (i.e., nonemulated) LANs, and may be interconnected with routers. LANE easily supports virtual networks over a single ATM network, while also offering the advantage of easy reconfiguration.

In LANE the point-to-point ATM switch provides the function of a virtual shared medium. From the protocol stack's point of view, the

ATM layer behaves like an IEEE 802 MAC protocol underlying the LLC. The key attribute of the shared medium connection is that communication is done as a broadcast. Every station in a LAN receives all the packets from all other stations, and filters out the packets destined to itself. This feature of broadcast can be emulated in ATM networks using broadcast servers even though ATM is originally connection-oriented.

LANE provides communication of user data frames among all its users, similar to a physical LAN. The communication channel between nodes on the same LANE consists of direct ATM connections between the nodes. Each LAN is an emulated entity on an ATM network based on the configuration data put into the LANE servers. There is no direct mapping between an emulated LAN and physical boundaries within a single ATM network. This means that there can be one or more *emulated LAN* (ELAN) running on the same ATM network. Each of the ELANs is logically independent, with the nodes connected to one ELAN being unable to directly communicate with the nodes connected to another ELAN. Any type of communication between ELANs requires some type of interconnection devices such as bridges and routers. This directly mirrors the characteristics of the real world.

The fact that a number of ELANs may run on the same ATM network is an important advantage of ATM networks. It enables the configuration and operation of virtual LANs. This was possible even before the *virtual LAN* (VLAN) specifications were defined by the IEEE [6, 7]. Emulating physical LANs also mean that LANE must have some other important characteristics. One is that it must be able to support connectionless services. That is, a sender must be able to send data without previously establishing a connection, which is a big problem for connection-oriented ATM networks. Additionally, LANE must support multicast, more specifically, the use of multicast MAC addresses (e.g., broadcast, group, or functional MAC addresses). This puts some constraints on the MAC driver interfaces in ATM stations. By supporting such characteristics, LANE enables existing applications to use an ATM network through existing protocol stacks and APIs such as IP, IPX, *advanced peer-to-peer networking* (APPN), NetBios, and AppleTalk.[7]

7. Since LANE emulates the IEEE 802 MAC layer below the LLC it can support not only IP but also various network layer protocols such as SNA/APPN, IPX, and NetBios. This contrasts the classical IPoA approach, which can support the IP suite only. While LAN emulation is capable of supporting many different network layer protocols, today the main protocols supported are IP and IPX, with IP becoming the de facto standard network protocol due to the popularity of the Internet.

ELANs enable configuration of multiple, separate domains within a single ATM network. The resulting configuration would be logically analogous to a group of LAN stations attached to an Ethernet/IEEE 802.3 or 802.5 LAN segment. Several ELANs could be configured within an ATM network, regardless of the physical location of each connected end system. An end system may belong to multiple ELANs, where each individual ELAN is logically independent.

However, the LANE has expansion limitations. While clients in an ELAN can communicate with each other directly, communications between ELANs are possible only through bridges or routers. Accordingly, LANE is suitable for small workgroup networks in a local area. ELANs interconnected by bridges and extended beyond a local area or small number of workgroups would be impractical. As the number of connected ELANs grows, the broadcast traffic passing over the bridge increases, and the bridge could become a bottleneck. To reduce the broadcast traffic, routers can be used instead of bridges, and to reduce the number of interconnection devices, multiple ELANs can be interconnected by direct ATM connections.

7.1.3.2 LANE Elements

LANE service architecture is based on a client-server model. Figure 7.6(a) shows the elements of a LANE-based network along with the connections that are used, and Figure 7.6(b) shows the protocol stack of LANE.[8]

As shown in Figure 7.6(a) the basic servers used in a LANE network are a *LANE server* (LES), a *LANE configuration server* (LECS), and a *broadcast and unknown server* (BUS). Broadly speaking, the LES is responsible for registering MAC addresses to ATM addresses and resolving the addresses. The LECS locates the LES and provides configuration information for each ELAN segment. The BUS delivers broadcast or multicast frames, and is responsible for delivering the unicast frames whose destination address is either unregistered or unresolved yet. Figure 7.6(a) also shows the multiple relationships and the types of connections between these servers and clients. The communication between the LECs and the communication between the LECs and the LE service servers are carried out over ATM VCCs. Each LEC communicates with the LE service servers over control VCCs.

8. Note that the architecture in Figure 7.6 is only functional, so LANE service configurations are not necessarily implemented in these three parts physically. For example, an LES and a BUS may be colocated on the same ATM switch, while LECS may be configured on a separate stand alone server so that it can function as a server for multiple LES/BUS ATM switches.

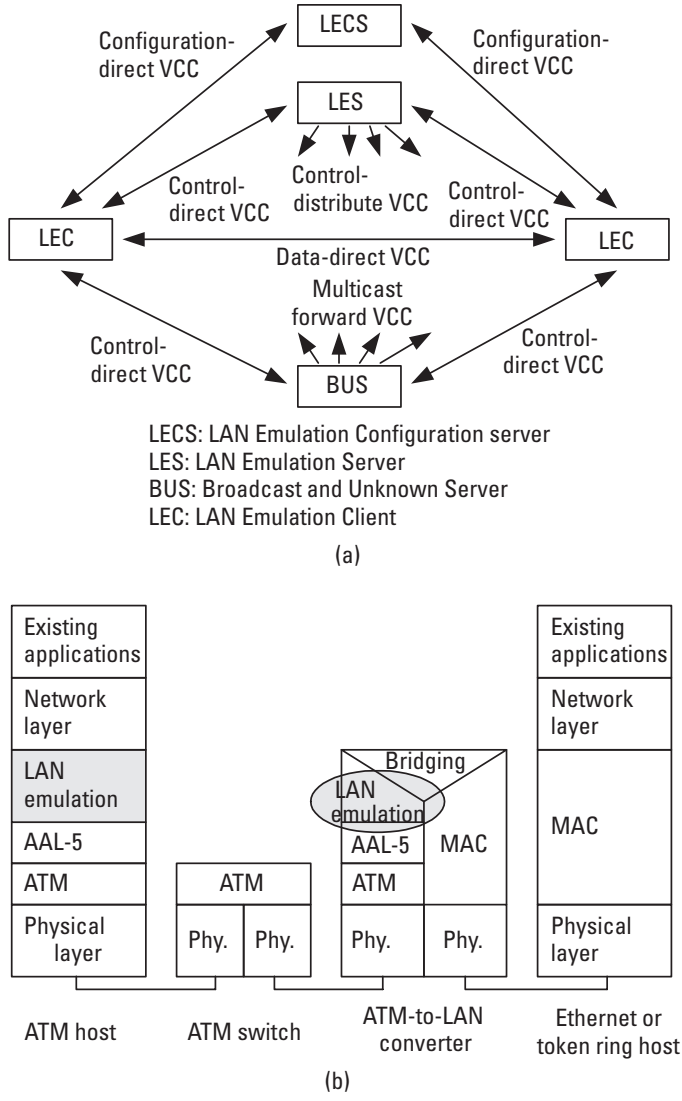

Figure 7.6 LANE: (a) elements and connections, and (b) protocol stack.

A single ELAN is a set of *LANE clients* (LE clients, or LECs) receiving *LANE service* (LE service) from a single group of servers. The LE service is offered by a group consisting of one or more LECS, one or more LES, and one or more BUS server. Conceptually this is shown in Figure 7.7. In the

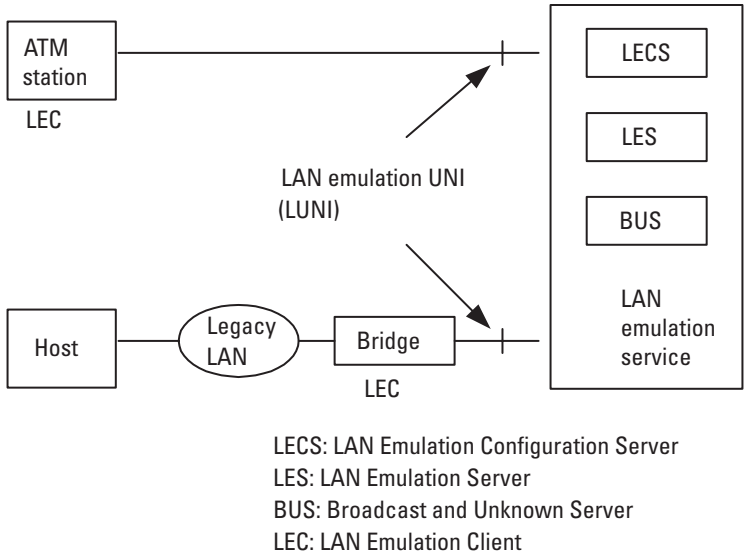

Figure 7.7 ELAN configuration and LUNI.

figure, *LANE UNI* (LUNI) defines the basic interface between an LEC and the servers providing the LE service. The LEC may be an ATM station, a LAN bridge, or a router with ATM interface. As shown in Figure 7.7, it is possible to connect a legacy LAN and an ELAN by using a layer 2 bridge. Note that in the same configuration it is also possible to use a layer 3 router and connect the two LANs at the layer 3 level.

LEC. An LEC is basically an ATM end station with an ATM address. As an LEC emulates a LAN node it is also assigned a MAC address, on the basis of which an LEC provides a MAC level emulated Ethernet/IEEE 802.3 or IEEE 802.5 service interface to the upper-layer applications. Any ATM end systems, for example, ATM workstations and ATM bridges, may be configured to be an LEC. It only needs to have the appropriate software and be configured with a method for finding the initial LECS to which it must connect. An LEC must implement the LUNI interface in order to communicate with other components within a single ELAN.

LES. An LES is the key control point of an ELAN, offering the control coordination function for the ELAN. It is basically a facility for registering

and resolving unicast and multicast MAC addresses and/or route descriptors to ATM addresses. An LEC must be connected to only one LES. The LEC must register its MAC address with this LES. It is possible for an LEC to register multicast MAC addresses with the LES and thereby function as multiple LECs on a single ELAN.

An LEC will also query its LES to resolve a MAC address and/or route descriptor to an ATM address. That is, the LES takes on the responsibility of emulating the ARP functionality that is normally used in Ethernet or token ring networks to resolve MAC addresses. The LES will either respond directly to the LECs or forward the query to other servers. In such a case the LES is acting as a type of proxy server.

BUS. A BUS server provides a multicast server function to provide multicast connectionless data delivery. A BUS server basically handles the data sent by LEC to the *broadcast* MAC address. It is a server designed to emulate the broadcast ability that both Ethernet and token ring LANs have but that ATM networks do not. In addition, a BUS server is used to transmit initial unicast data sent by an LEC before the data-direct target ATM address has been resolved and, consequently, before a data-direct VCC has been established. Since the direct VCC has not been established, the data is sent on the multicast VCC to ensure that it is broadcast to all LECs on the same ELAN, as the packets are broadcast to all other hosts on the LAN. The BUS server also participates in the *LE address resolution protocol* (LE-ARP) to enable an LEC to locate its BUS.

A BUS sees all broadcast, multicast, and unknown traffic to and from an LEC and distributes data with multicast MAC addresses (e.g., group, broadcast, and functional addresses) to all the LECs it is connected to. An LEC is configured to always see only a single BUS server. If an LEC does not need to receive all multicast MAC addressed frames, the BUS server may then selectively forward multicast MAC-addressed frames to only those LECs that have requested them. To ensure that AAL-5 frames from different sources are not interleaved the BUS server must implement a serialization function in transmitting cells from different clients. Some LECs take advantage of the multiple interfaces of the BUS and send frames destined to a specific multicast MAC address to a different BUS interface.

LECS. An LECS is the main configuration server for an ELAN. An LECS assigns individual LECs to different ELANs. As such it must be configured beforehand along with the information regarding which LE clients are to be assigned to which ELAN. An LECS assigns any client to a particular ELAN

service by giving that client the appropriate LES ATM address. The LEC then contacts that LES and thereafter operates as a member of the ELAN that is served by that particular LES. An LECS may assign a client to an ELAN based on either the physical location (ATM address) or the identity of a LAN destination (i.e., MAC address) that it is representing.

All LECs must be able to obtain information from an LECS using the configuration protocol. During the initial boot-up, an LEC must first contact its LECS. The LEC must either be preconfigured with the address of the LECS or may get it through other methods such as *interim local management interface* (ILMI).

Types of VCCs used in LANE. LANE uses a number of different types of VCCs to operate correctly. Broadly speaking, LANE uses two types of VCCs—point-to-point VCCs and point-to-multipoint VCCs. The *configuration-direct VCC*, *control-direct VCC*, and *data-direct VCC* belong to the point-to-point VCC category; and the *control-distribute VCC*, *multicast send VCC*, and *multicast forward VCC* belong to the point-to-multipoint VCC category.

Point-to-point VCCs. A *configuration-direct VCC* is a bidirectional point-to-point VCC that is used by the LEC to exchange configuration messages with the LECS. This VCC must be established during the initialization phase of operation. The LEC uses this VCC to receive configuration information from the LECS. The LEC and LECS may release the configuration direct VCC once the LEC is connected with the LES.

A *control-direct* VCC is a bidirectional point-to-point VCC that is used by the LEC to send control traffic to the LES. This VCC must be established during the initialization phase of operation. The LEC is required to accept control traffic from this flow. The LEC and LES must not release the control-direct VCC while participating in the ELAN.

A *data-direct VCC* connects the LECs with each other. It is established by the LEC once it knows the ATM address of the destination node by address resolution. The data-direct VCC is used to carry encapsulated Ethernet/IEEE 802.3 or IEEE 802.5 data frames. This VCC never carries control traffic.

Point-to-multipoint VCCs. A *control-distribute VCC* is a unidirectional, point-to-multipoint VCC that the LES may optionally establish for distributing control traffic to one or more LECs. This VCC may be set up by the LES as part of the initialization phase. If the control-distribute VCC is set up, the LE

client is required to accept the control-distribute VCC. The LEC and LES must not release the control-distribute VCC while participating in the ELAN.

A *multicast-send VCC* is a bidirectional VCC that connects the LECs to the BUS. It is used to transmit multicast data frames from the LEC to the BUS. Additionally it is used to transmit data frames for the destinations whose ATM addresses the LEC does not know. This is frequently the case when the initial data frame for a destination is received and the LE_ARP reply from the LES has not been received. An LEC may also receive multicast frames over this VCC.

A *multicast-forward VCC* is a unidirectional point-to-point or point-to-multipoint VCC established between the BUS and the LEC. It is used by the BUS to forward multicast data frames to the connected LECs. The multicast-send VCC and multicast-forward VCC are used to carry encapsulated Ethernet/IEEE 802.3 or IEEE 802.5 data frames. This VCC never carries control traffic.

7.1.3.3 LANE Procedures

The LANE service function uses the procedure consisting of *initialization*, *registration*, *address resolution*, and *data transfer*. At first, a client contacts the LECS to locate the LES. Then an ATM connection is established to the LES, and the client registers its MAC address and ATM address to the LES. These functions are defined over a LUNI.

Initialization and configuration. The overall flow of the initialization and configuration flow in LANE is shown in Figure 7.8. The first process for joining an ELAN involves getting some basic initial parameters and fetching the LECS's address. Then the client determines which ELAN it is connected to and fetches configuration information and the LES's address. Next, the join phase takes place where the client sets up a VCC with the LES and joins the ELAN.

During the initial configuration phase, a connection with the LECS must be set up. Specifically, a configuration-direct VCC must be set up between the LEC and the LECS. The address of the LECS server can be found by a number of different ways. It can be found either by manual configuration of the information in the client, by using ILMI signaling data from the nearest ATM switch, or by using a well-known VCI (VPI = 0, VCI = 17) to connect with the LECS directly. By connecting with the LECS, the LEC obtains the address of the LES and other configuration parameters.

It is also possible for the LEC to skip this LECS connection phase and not set up a configuration-direct VCC but, instead, directly connect to an

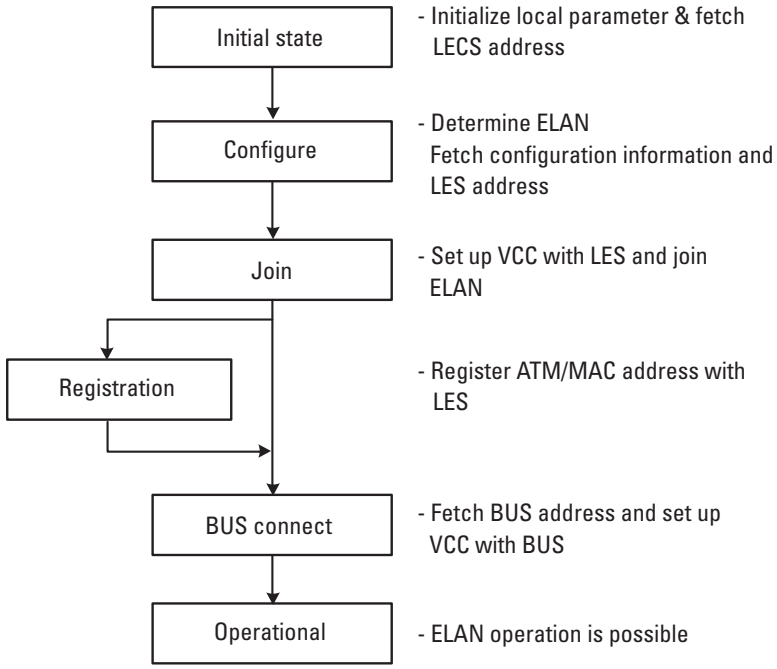

Figure 7.8 Initialization and configuration flow in LANE.

LES. For this to be possible the LEC must know the LES and other relevant information beforehand. How to find the address of the LES is not defined in this case. Manual configuration would be the most likely method.

The next phase following the configuration phase is the join phase, where the LEC establishes a control-direct VCC with the LES and attempts to join the ELAN with the name received during the configuration phase. Following this the client must register its MAC/ATM address with the LES. This is carried out by using the registration procedures to be described next.

Once the LEC is connected to the LES, the BUS connect phase begins. In this phase the LEC must connect with the BUS. This is done by first requesting the BUS's ATM address from the LES by using the LE-ARP function. The LEC sends an OK message to the LES and the LES responds with an OK message containing the BUS's ATM address. The LEC then establishes a multicast-send VCC with the BUS server. The BUS adds the LEC to a multicast forward VCC. The LEC is then connected to an ELAN and may start to transmit data.

Registration and deregistration. The registration procedure is used initially by the LEC to register its MAC/ATM addresses with the LES during the configuration and join phase. The procedures may be repeated to register extra MAC/ATM addresses. Deregistration is the opposite process by which previously registered MAC/ATM addresses are removed from the LES. As shown, clients may register multiple MAC addresses by using the registration procedure repeatedly. Bridges that want to register the MAC addresses of clients that are not LANE capable can also use such functionality. Such a case would apply to the bridge that hides multiple hosts on the legacy LAN segment as shown in Figure 7.7.

Address resolution. In LANE, each client maintains information on mapping MAC addresses to ATM addresses for connection setup. When a client does not have this information, it can obtain ATM addresses from MAC addresses using the LES. The LES makes a table for mapping MAC addresses to ATM addresses at the registration stage and updates it whenever change occurs.

The LEC obtains the ATM address by sending a request message to the LES. Then the LES normally replies with a response that contains the appropriate ATM address. The request message contains the source MAC address, the source ATM address, and the destination MAC address. The response message contains the destination ATM address in addition to the information in the request message. The request message is sent over the control-direct VCC, but the response message can be sent over either the control-direct VCC or the control-distribute VCC. The advantage of the sending the response message over the latter is that the response may be received by all the LECs connected to the control-distribute VCC. The other LECs may then cache the information for future uses.

There are three different operation scenarios depending on the type of MAC address and whether the destination MAC address has been registered with the LES. First, if the MAC address is a broadcast MAC address, the LES returns the address of the BUS. Second, if the MAC address is registered with the LES, then the LES will reply with the ATM address corresponding to the MAC address. Third, if the MAC address is not registered with the LES, then the LES forwards the request to all the LECs.

Data transfer. In LANE, data transfer may be done by either unicast or multicast frame. Once the MAC address to ATM address resolution is completed, the client begins the connection setup process. The connection setup at this time directly between the LECs is the data-direct VCC. After this process, unicast frames are transmitted directly to the destination client. The

connection is automatically released after a fixed length of idle time. When transmitting multicast frames, the client takes a slightly different procedure: In this case the client transmits the multicast frame to the BUS using the multicast send VCC. The BUS then forwards this frame to all the other LECs by using the multicast forward VCC.

A client can also use the BUS to send frames before the address resolution process gets finished. By doing so the delay between transmitting a frame and its reception by a destination is minimized. In this case the BUS broadcasts the frames to all clients, because it is the only method available to send frames to destination clients with unresolved MAC addresses. In addition, for a delay-sensitive application, it is desirable to send frames using this method so that the service is unaffected by the delay that may occur during the address resolution and connection setup process. Once the address resolution procedure succeeds, a data-direct VCC is set up and data is no longer forwarded through the BUS server. To prevent the clients from abusing the broadcast channel, the number of broadcast frames that a client may send within a given time period is limited.

If the BUS were capable of delivering the unresolved frames only to their destinations without broadcasting, then the traffic in the network would be much reduced. However, it would result in complicated and costly BUS implementations.

7.1.3.4 LANE Implementation

The LANE specifications only define the functions and interfaces for LANE operation. From a practical point of view, the actual implementation details are left to the implementer. Typically, LECs are implemented in ATM end stations, either as a software driver or as an ATM adapter (i.e., ATM-specific hardware and software). ATM end systems can be either intermediate systems (e.g., bridges or routers) or end stations (e.g., hosts or PCs). The LE service might be implemented in any combination of ATM switches and ATM attached end stations (e.g., bridges, routers or workstations). An LE service component may be colocated with an LEC. Note that it is important that an LES can also be located in an ATM end station as the server communicates with the LECs and other servers by using LANE.

7.1.3.5 LANE Versions

Currently two versions of LANE have been defined. LANE version 1.0 defined only the LUNI specifications and operations as described above. There were a number of limitations in the initial standards: Only one LES

could be defined for an ELAN making it a bottleneck and a potential single point of failure. Also it only supported UBR services, thereby rendering QoS-based services impossible.

LANE version 2.0 was later defined. It was comprised of LUNI version 2.0 and *LANE NNI* (LNNI) specifications. The former defines the interfaces between the LESs and LECs and focuses on the definition of LANE operation in a single emulated LAN, whereas the latter defines the interfaces between the LECS, LES, and BUS servers in the LE service and focuses on the definition of LANE operation when multiple LANE servers are involved. One of the main problems with LANE version 1.0 is the fact that as only a single LES can be used, there is a single point of failure. By defining interfaces between the multiple LESs and thereby enabling the use of multiple servers this problem can be eliminated in LANE version 2.0. The specifications allow for up to a maximum of 20 LESs and 20 BUSs. LANE version 2.0 also adds support for globally and locally administered QoS, enhanced multicast with selective broadcasting capability, and support for ABR rate-based flow control, and support for FDDI. Also added are support for LLC-multiplexed VCC and support for MPOA.

The enhanced multicast defined in LANE version 2.0 adds support for separating multicast traffic from the general broadcast path. It enables the possibility of determining which members of the emulated LAN are to receive multicast frames. The filtering function is performed through cooperation between the source and the LANE service.

7.1.4 MPOA

Multiprotocol over ATM (MPOA) was developed by the ATM Forum as a comprehensive solution for interconnecting various types of layer 3 networks by using ATM technologies. Where LANE solved the problem of using ATM to emulate LANs, MPOA aimed to efficiently use ATM to support the interconnection of LANs, both emulated and nonemulated.

The main goal of MPOA was to efficiently support internetwork traffic using ATM technology. The key idea was to use the direct ATM connections between end nodes where possible. This is similar to the concept of LANE, except that it is now extended to include the nodes not only in the same ELAN, but also in other LANs. This is illustrated in Figure 7.9. By using the protocols defined in MPOA along with other protocols such as NHRP, it is possible to set up a direct ATM connection to any other systems connected to the same ATM network.

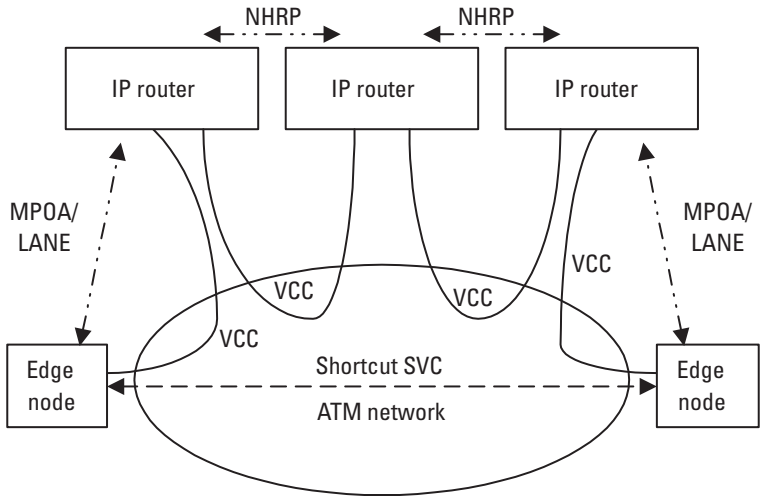

Figure 7.9 Use of MPOA to set up a shortcut SVC over an ATM network.

The method chosen for solving the above problem was to provide the functionality of a router over an ATM network using a distributed model and implementing it in a number of switches and servers in the network, along with the corresponding changes in the client software. It is a distributed model with the relevant functionalities distributed over the network in the form of server-client functions. At the same time it offers a single paradigm for overlaying internetwork layer protocols over ATM.

The characteristics of the solution include the efficient transfer of intersubnet unicast data in a LANE environment. It also provides a scaleable routing solution for IP networks by operating as a distributed virtual router in ATM networks. The routing functions are distributed over various physical boxes. While basic routing protocols such as OSPF and RIP are run on a server, IP data forwarding is carried out by ATM switches. For nodes in the same network, LANE is used to support data transfer in subnets, while the NHRP is used to support address translation between different subnets so as to enable the setup of connections between them.

7.1.4.1 Virtual Router Concept

One of the key ideas in MPOA is the concept of the virtual router. This concept is built on the idea of physically dividing the routing and forwarding functionalities in the internetwork layer; routing protocol and route

computation are handled by separate router servers, while the actual packet forwarding is carried out by the ATM switches and edge devices. This can be construed as an example of separating the intelligent control function from relatively simple forwarding functions. This is a radical departure from the current generic router structure. The traditional single-box router contains both of the above functionalities in one box. It must run complicated routing protocols at the same time, also maintaining a fast path for forwarding user packets.

As the network speed and capacity increase, the routers must be speeded up at the same time. This may not be easily done in conventional structure, but in a virtual router structure, the servers and forwarding engines may be upgraded separately. Figure 7.10 helps to visualize what this means. It compares the structure of a single-box switch with that of an MPOA virtual router. A single-box switch may be roughly divided into a central processor, I/O cards, and a backplane. The central processor is the main controller of the router with the main responsibility for carrying out any route calculations. The I/O part classifies and forwards packets. The backplane interconnects the I/O ports and forms the path between the I/O ports (refer to Section 3.3.3 for more details). In a virtual router, these components can all be implemented in separate boxes. That is the central processor implemented as an MPOA server, the I/O ports as MPOA edge devices, and the backplane as the ATM-switched network that interconnects all the MPOA servers and edge devices.

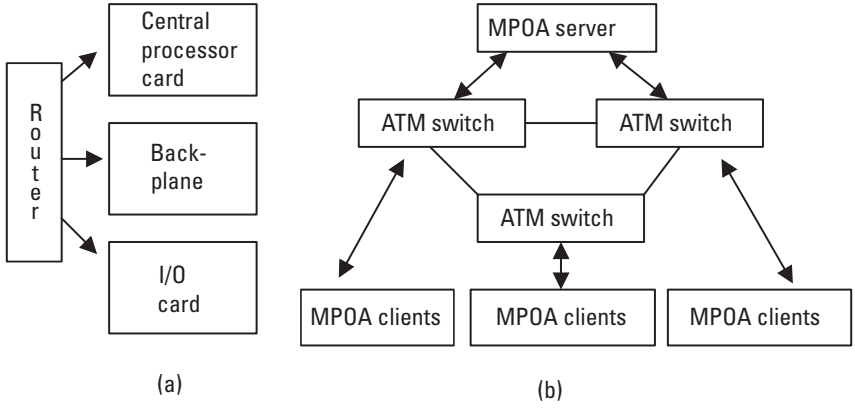

Figure 7.10 Comparison between (a) traditional router structure, and (b) MPOA-based virtual router structure.

Due to the specialized nature of each node, the virtual router will be easier to upgrade. For example, increasing the capacity of the ATM switch network may be done simply by adding more switches or by upgrading the switches themselves. Running more complicated control processing may be managed by increasing the number of servers. In contrast, increasing the speed or the processing capacity of a traditional router system requires increasing the routers themselves.

In addition, it is harder to introduce new routing functionality to a single-box router than to a virtual router. The idea of separating the routing functionality from the actual packet-forwarding component is also the main basis of the MPLS paradigm. (Refer to Section 3.3.5 for more detailed discussions on other benefits of such a separation.)

7.1.4.2 MPOA Elements

There are two main MPOA elements: one is *MPOA client* (MPC), and the other is *MPOA server* (MPS). An MPOA edge device contains an MPC and a LANE client. An MPS contains NHS servers and routing functions (and also a LANE client), as shown in Figure 7.11.

MPOA client (MPC) functions. The main function of the MPC is to support data transport by using a layer 3 short-cut. MPC only does layer 3 forwarding, not routing. The MPOA client uses an NHRP-based MPOA request/response protocol for short-cut transport. Once configured, the short-cut information is stored in an ingress cache table. Frames received over the short-cut path are passed to upper layers after appropriate data link layer encapsulation.

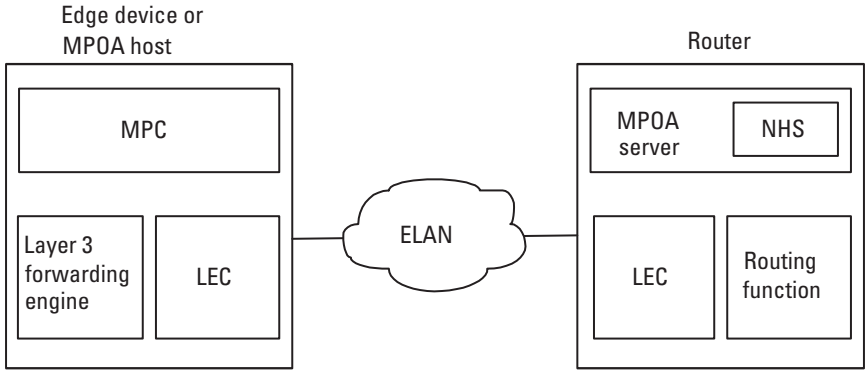

Figure 7.11 MPOA elements.

MPOA server (MPS) functions. The main function of the MPS is to implement the logical structure of a router. The MPS supplies to the MPC the layer 3 forwarding information learned by mapping the layer 3 address with the ATM address. It interacts with the local NHS routing functions to give information to the MPC. In addition, the MPS changes the MPOA request/response to an NHRP request/response for the MPC. Routers also run routing protocols such as OSPF, RIP, and IS-IS. An NHS is used to provide address resolution between ELANs.

Internetwork address subgroup (IASG). An important concept to understand in MPOA is the concept of IASG. An IASG is an address group that contains a wide range of internetwork addresses that can be summarized and used in an internetworking layer routing protocol. An example of an IASG is the IP network layer prefix.

7.1.4.3 MPOA Operation

MPOA consists of a number of basic operational procedures such as configuration of the various MPOA components including the MPOA clients and MPOA servers, discovery of the MPOA servers and their addresses in a subnet, resolution of the ATM address of the destination, connection management of the various control and data connections, and the actual data transfer. In general, one can view LANE services as providing the configuration, discovery, and legacy LAN support, while NHRP and LANE together provide support for destination address resolution.

Configuration and discovery. The configuration and discovery procedure aims to get configuration information to MPOA clients. The information is mainly on the MPOA servers to which the client may be connected. Note that as in all cases it is possible to configure all the relevant information manually on a client, but this is not a scalable method.

The configuration information consists of the ATM addresses of the servers and some information related to the IASG that the MPOA server supports. This information would include the IASG identifier and protocols, the name of the ELAN, and the IASG address and prefix. As MPOA is defined to support a number of protocols besides IP, the IASG configuration is more generic than would be if it had been designed for just to support only IP. The basic configuration and discovery procedures are based on the LANE configuration procedures and methods. The ATM address of the MPOA server can either be configured into the client or the client may use ILMI to retrieve it from the nearest ATM switch.

Address registration. The MPOA address resolution method is based on the NHRP resolution scheme. In the NHRP scheme the NHS is expected to have a database containing the mappings between the destination ATM addresses and the destination IP addresses. The NHS replies to the resolution requests by checking this database. In a similar manner, the MPS must also maintain such a database. Consequently, the MPC must register the IASG addresses that it supports with the MPS.

Address resolution. The address resolution procedures are the methods by which target addresses are mapped to ATM addresses. Depending on whether or not the destination is in the same internetwork layer subnet, the method used for destination address resolution differs. If the destination is on the same internetwork layer subnet then the destinations can communicate by using a LANE connection or by using a bridge with a LANE interface. In such a case the address resolution is carried out by using the mechanisms defined in LANE (refer to Section 7.1.3). If the destination is on a different internetwork subnet then the destinations can be normally reached only by going through a layer 3 router. In such a case the NHRP protocol must be used to resolve the ATM address of the destination node or its nearest exit router (refer to Section 7.1.2).

Data forwarding. MPOA data transport procedures can be divided into that for the default transport case and that for the short-cut transport case. The default transport method is to use LANE, so consequently all MPOA elements, MPCs, and MPSs must support LANE. The key edge device is the layer 2 bridge. However, in the short-cut transport method, a path is set up by using NHRP-based address resolution and cache management mechanisms. In this case the key edge device is the layer 3 router. All transport inside a single ELAN uses only the default transfer and all transport between ELANs can use either the default transfer or short-cut transfer.

The data forwarding operation by an MPOA client follows the following steps: When a frame arrives at the MPC it is first forwarded along the default routed path. This means that it is forwarded by using LANE either to the destination or to the default router if the destination is not reachable by LANE. As more packets arrive, the MPC must decide whether the flow of packets warrants a short-cut path setup. If this is the case, MPOA address resolution is carried out to find the ATM address of the destination to which a short-cut path must be set up. The method short-cut path is then set up and data transfer may start.

Cache management functions. One of the functions in MPOA is the cache management function used to support short-cut connections. All MPCs must keep an *ingress* cache and an *egress* cache. A simplified example of the ingress and egress cache tables is given in Figure 7.12.[9] The ingress cache maps a combination of the MPS's MAC address and the destination IP address to the correct ATM connection. Additionally, it maintains a count entry that is used to decide whether the connection has enough traffic to warrant a short-cut path setup.

When the first packet of a flow first arrives at an MPC, the ingress cache table is checked for an entry for this packet's flow. If there is a cache miss, a new entry is made with the information available. At this stage ATM connection entry is empty. The count entry is set to 1 and incremented with each packet transmitted. If more than a predefined number of packets are transmitted within a certain period of time then a short-cut path is setup for the flow. The VPI/VCI of the short-cut path is put in the ATM connection entry. Thereafter, any packet of this flow is automatically transmitted by that ATM connection. The information for updating the ingress cache (i.e., VPI/VCI of the ATM connection) is updated when the MPC receives the MPOA resolution response. If the packet flow is idle for a certain amount of time the cache entry is removed.

The egress cache maps a combination of the incoming ATM connection and the destination IP address to the correct MAC address and port

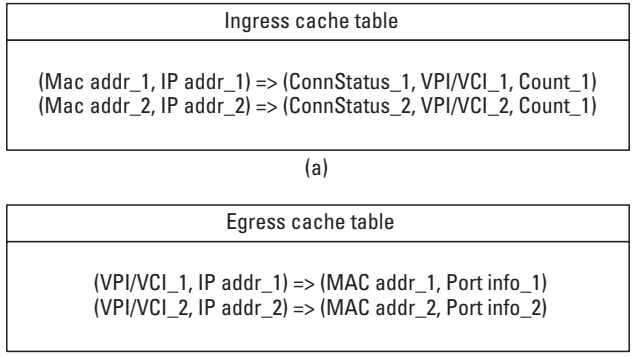

Figure 7.12 Example of the caches in MPOA: (a) ingress cache, and (b) egress cache.

9. Note that the exact cache entries and forms may differ from implementation to implementation.

information. When the egress MPC receives a packet, if a corresponding entry can be found in the egress cache table, the MAC address is appended to the packet and the packet is sent out on the appropriate port. The egress cache entries are updated when an MPOA egress cache imposition request is received from the egress MPS.

7.1.4.4 MPOA Message Types

As explained in the previous section, MPOA is based on the LANE and NHRP protocols. As LANE version 2.0 was designed from the beginning to support MPOA, no additional messages or information types need to be added. However, to support some extra functionalities of MPOA, a number of extra messages and information elements have to be added to the base NHRP protocol. All such messages follow the basic NHRP format. The *initial resolution request* is sent by the (ingress) MPC to the (ingress) MPS to request that the ATM address for an IASG address be resolved. The *resolution reply* is sent by the (ingress) MPS to the (ingress) MPC and contains the resolved ATM address. The *cache imposition request* is sent by the (egress) MPS to the (egress) MPC, instructing it to set up an egress cache entry for the connection. The messages needed for correct operation of the protocol include egress cache purge, keep alive, trigger, and data plane purge.

7.1.4.5 Example of MPOA Operation

The steps shown above give a rough idea how MPOA operates. A more detailed explanation on MPOA operation can be found in the sequential steps given below. Figure 7.13 illustrates the basic operation of MPOA.

- Step 1: The source transmits a packet to the destination host. The first packet arrives at the MPOA edge device (C1) on its legacy LAN interface. On checking the ingress cache for an entry, there occurs a cache miss indicating that this packet is a new flow. A new ingress cache entry is made and the packet is forwarded to the next hop router by way of the ELAN—that is, the first packets are forwarded along the default router path.
- Step 2: As further packets arrive, the ingress cache entries are updated. After a while, the MPOA client decides that the flow is sufficiently long-lived to warrant a new connection to be set up.
- Step 3: The MPOA client sends an MPOA resolution request to the nearest MPOA server (S1). (The MPOA client is configured with the address of the nearest MPOA server beforehand.)

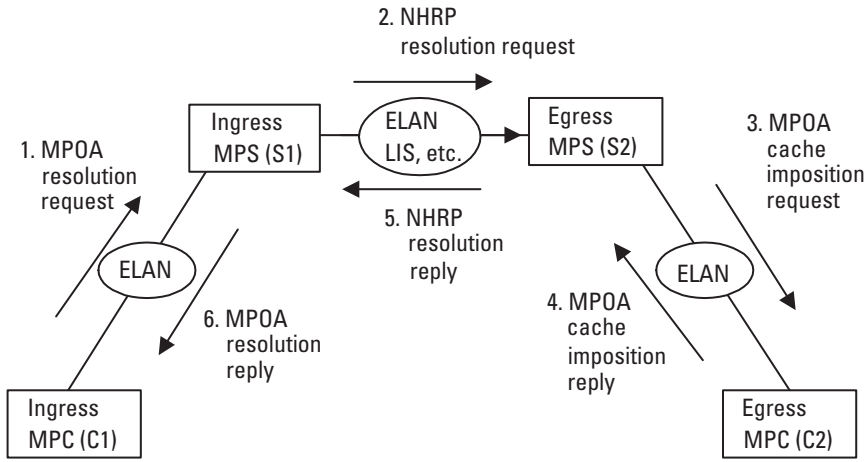

Figure 7.13 Basic operation of MPOA.

- Step 4: The MPOA server transforms the MPOA resolution request message into an NHRP resolution request and forwards it to the appropriate next hop router according to the internetwork destination.

- Step 5: The NHRP resolution request eventually arrives at an MPOA server (S2) that knows how to directly reach the destination. Usually this means that the MPOA server is on the same emulated LAN as the MPOA client.

- Step 6: This last MPOA server locates the MPOA client (C2) to which the destination is directly connected and sends a cache imposition request to the MPOA client. The client creates an egress cache entry for this flow.

- Step 7: The MPOA client (C2) generates a cache imposition reply that is sent back to the MPOA server.

- Step 8: The MPOA server (S2) receives the cache imposition reply, transforms it into an NHRP resolution reply, and sends it back to the MPOA server that originally generated the request. The reply follows the path that the original NHS resolution request took when coming to the destination (refer to Section 7.1.2).

- Step 9: The first MPOA server (S1) receives the NHRP registration reply and relays this back to the first MPOA client (C1) as an MPOA resolution reply.
- Step 10: The MPOA client (C1) now has enough information to set up a direct ATM SVC connection with the final MPOA client (C2). Once this connection is set up data is forwarded along this connection directly to the final MPOA client, instead of going through the complicated default router path.

7.1.5 I-PNNI

I-PNNI is an attempt to extend PNNI to support the integrated model for IP and ATM interoperability. As explained in the introduction to this chapter, the previous methods (classical IPoA, LANE, NHRP, MPOA) are all overlay network model–based architectures. The IP nodes at the edges are completely unaware of the underlying ATM network structure. In contrast, I-PNNI results in the IP nodes becoming aware of the ATM network's topology. The ATM nodes and the IP nodes are peers communicating control messages and signals to each other.

By using I-PNNI both ATM and IP nodes become fully aware of each other's network topology. This is achieved by using common (or interoperable) routing protocols and addressing schemes. More specifically, I-PNNI achieves these aims of supporting both ATM and IP by using a single routing protocol based on link state routing protocol. I-PNNI supplies basic information needed for SVC setup to both ATM and IP networks. Originally it was planned to be an extension of PNNI-based ATM switch functionality. I-PNNI is applicable to ATM and non-ATM (or datagram) media. It can be used between ATM switches, between routers, and between ATM switches and routers.

The basic flow of operation that was originally envisioned by its proponents is as follows: I-PNNI nodes exchange hello messages with neighbors to determine local topology. Nodes then announce their topology, with the announcements being flooded throughout the area. Based on this information, nodes are able to understand complete topology of the area and calculate routes.

As yet the protocol is not completely developed and no further active work has been carried out lately. It has now been mostly supplanted by MPLS as the means to integrate ATM and IP networks with a single routing technology.

7.1.6 MPLS over ATM

As explained previously in Chapter 3, MPLS is conceptually a combination of layer 3 routing and layer 2 switching. It employs level 3 routing protocol functions, level 3 forwarding at the edge nodes, and level 2 forwarding in the core nodes. By combining the IP control paradigm with the label-switching–based traffic forwarding, it is possible to get higher bandwidth, enable traffic engineering through explicit routing mechanisms, and garner various other advantages. (For a more detailed explanation on the principles and practical aspects of MPLS, refer to Section 3.3.5.)

As noted above MPLS has many advantages, including that MPLS enables the conversion of ATM switches into IP routers. This means that cheap ATM switches may be converted into high-performance IP routers by applying minimal changes. This can be easily understood if one recalls the relationship between MPLS and ATM.

Currently, MPLS in general is being designed to explicitly allow *label switch routers* (LSRs) to be based on traditional router platforms and on ATM switch platforms as well. The reason for the emphasis on ATM platforms is a natural progression from the fact that ATM networks pioneered many of the basic underlying building blocks that constitute MPLS. These include the concepts of simplified forwarding, traffic engineering based on explicit routing, and QoS routing. Any MPLS devices must inevitably incorporate these building blocks and ATM devices, which already dealt with these features within ATM networks, can support them very easily.

A simplified view on how an ATM switch functions as an LSR is as follows. The labels of an MPLS packet are carried in the VPI/VCI fields of the ATM cell. Conceptually, an ATM VPI/VCI header is in fact equivalent in semantics to an MPLS label. The operation of an MPLS LSR on the MPLS label mirrors the action that the ATM switch carries out on the VPI/VCI header of the ATM cell. The VPI/VCI header fields of an ATM cell may be used as a single label or it may be divided up into two different labels. In other words, the VPI/VCI fields may be viewed as a label stack, with the VPI used as the top label in the stack and the VCI used as the second label.

While the mapping of MPLS functionality to ATM network hardware is conceptually simple, it is not a completely cut and dry affair. There are a number of issues that must be solved before it becomes possible to use MPLS control architecture over ATM network hardware. We examine a number of these issues in the following sections.

An ATM switch may have multiple interfaces, with some working to the original ATM specifications and some operating as MPLS interfaces. To

distinguish the two types, we may call an ATM interface that operates under the "MPLS label switch controlled" paradigm an LC-ATM. An LC-ATM operates in a different mode from the normal "conventional" ATM.

7.1.6.1 Basic Operation of an MPLS-ATM Switch

The MPLS-ATM switch preserves the ATM user plane. That is, user data is switched and transmitted in the ATM switch as in any other normal ATM switches. When an ATM cell is received its VPI/VCI is used for table lookup, based on which a new VPI/VCI label is assigned. Additionally traffic parameters and policing may be also carried out on the connection according to the data lookup.

The main difference between the MPLS-ATM switch and the ITU-T/ATM Forum-defined ATM switch is that the control plane of the MPLS-ATM switch mainly runs IP routing protocols such as OSPF, RIP, BGP, and PIM rather than UNI and PNNI protocols. The forwarding operation is determined by the underlying ATM switch fabric, whereas the control functionality is similar to that of a router. This is because cell forwarding is dictated by the switch fabric, while switch functionality is largely defined by the control component.

7.1.6.2 Encapsulation of Labeled Packets on ATM Links

As mentioned previously, the MPLS label is carried in the VPI/VCI field of the ATM cell. As the size of the VCI field is 16-bits-long, there are up to 2^{16} labels available. If the VPI field is used as well, additional 8 bits (or 12 bits) are available for the labeling. As the tag stack can have two layers, it is possible to make the VCI field define one layer and the VPI field the other.

There are two points to note in using MPLS over ATM networks. First, the label stack field header that is normally used by MPLS packets is also included in the ATM AAL-5 PDU in front of the actual network layer packet. The top level is carried in the ATM VPI/VCI while the lower layers are in the stack. The whole stack is carried, but the top level is ignored as it is already in the ATM VPI/VCI. This approach enables label stacks of arbitrary depth. As mentioned previously the use of the VPI/VCI fields in the ATM cell header essentially limits the label stack to only two layers. By including the whole stack in the payload, it is possible to support arbitrary depth label stacks. This approach also helps to solve the problem of transporting TTL and *experimental* (EXP) bits. ATM VPI/VCI fields do not have any fields for carrying the TTL and EXP bits that are in the labels defined in MPLS.

Second, remote binding is always acquired on demand. This means that an ATM LSR will not advertise its local binding to another ATM LSR before the other ATM LSR specifically requests the binding. In order to acquire remote bindings the ATM LSR must specifically request it from the other ATM LSR. By requiring that bindings are only acquired on demand, the number of bindings actually needed is minimized. This is helpful in practical situations because many commercial ATM switches have a built-in hardware limit in the VCs that they can support. Though the use of fixed labels (VPI/VCI fields) is helpful in simplifying hardware design, it does not necessarily mean that lookup tables based on 24 bits are feasible. Most ATM switches are designed to use a smaller portion of the VPI/VCI fields. On-demand remote binding of labels also helps solve the cell interleave problem to be discussed below.

7.1.6.3 Cell Interleave Problems

As discussed in Section 3.2.1, in IP routers, destination-based forwarding is normally used. This can lead to an interesting problem called the *cell interleave problem.* This problem occurs when multiple MPLS streams meet at an ATM switch and are all routed toward the same destination, as shown in Figure 7.14(a). As all the packets are to be sent to the same destination, they are all given the same MPLS label. In the case of ATM networks, this means that the cells from each stream are given the same VPI/VCI values. Consequently, the receiving host would have no way to discriminate the cells sent by source *A* from the cells sent by source *B* as they would all have the same VPI/VCI value. In such a case the receiver will not be able to correctly reassemble the packets from those cells as they would be mixed up.

One proposed solution to this problem is the so-called *VC-merge* method shown in Figure 7.14(b). In this scheme, VC-merge occurs at the ATM switch where the two streams meet. While the cells of the first stream are being transmitted, the ATM switch buffers the cells coming from the other stream. This is done until a full packet has been sent. The switch recognizes that a full packet has been sent by examining the end-of-frame bit in the ATM headers of the cells in the stream. This function is usually implemented in the switch to support frame-based discard methods for improving TCP/IP traffic performance. Once all the cells from one stream have been sent, then the cells from another packet are sent. This second packet may or may not be from the same stream. The key point is that it is ensured that cells belonging to different end user packets are not interleaved.

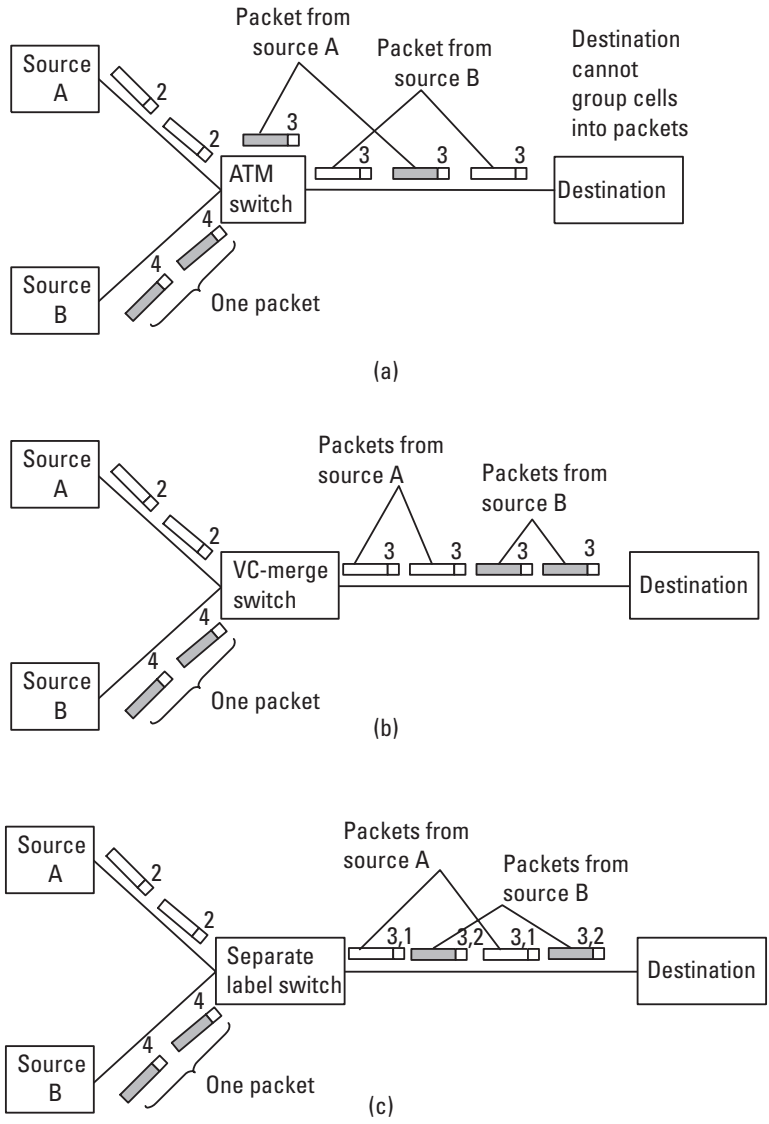

Figure 7.14 Illustration of the cell interleave problem in MPLS/ATM: (a) basic cell interleave problem, (b) VC-merge solution, and (c) separate label solution.

Another proposed solution is to use separate labels for different streams, as illustrated in Figure 7.14(c). As each stream is allocated a separate label, there is no problem at the receiving node in differentiating the

different streams. The price, however, is that a large strain is placed on the number of labels (i.e., VPI/VCI values) needed. Consequently, the on-demand method must be used for assigning the labels.

7.1.6.4 Looping and TTL Adjustment

Another problem with LC-ATM links is that when the packet is sent over an ATM link there is no TTL field in the ATM cell. As such, the problem of protecting against routing loops is an important issue. A solution may be found in applying the loop-mitigation method used in the MPLS to MPLS-ATM networks. This can be accomplished by having the TTL values decremented by the number of ATM LSRs that the path traverses. As mentioned above, ATM cells do not contain any TTL field and ATM LSRs themselves cannot look inside an ATM cell stream and decrement the TTL field. So, instead, the non-ATM LER at the starting point of the MPLS path decrements the TTL field by the number of the ATM LSRs that the path traverses. This number is learned during the path setup time from the label mappings. If the TTL value happens to become a negative number due to the decrement, the packet is handled in an appropriate manner.[10] Otherwise, the TTL value is decremented and put into the packet. When the packet comes out of the path at the end, the TTL value is then corrected.

7.1.7 CLSF-Based Data Services

The ITU-T recommends two different configurations to support connectionless data services in BISDN: the *indirect* and *direct* methods. In the indirect method, *connectionless service functions* (CLSFs) and the associated adaptation layer entities are located outside the BISDN, whereas in the direct method, CLSFs are located inside the BISDN. Figure 7.15 shows the connectionless service reference models for these two methods.

7.1.7.1 Indirect Method

In the indirect method, connectionless services are provided through the virtual connections that connect ATM *interworking unit* (IWU) pairs. Each IWU provides an interface between a connectionless LAN and the ATM network. Virtual connections connecting IWUs form a dense mesh of connections in the network. Such connections may be established all together by

10. For example, the packet may be dropped and an ICMP error message may be generated and sent without being mapped to a label path.

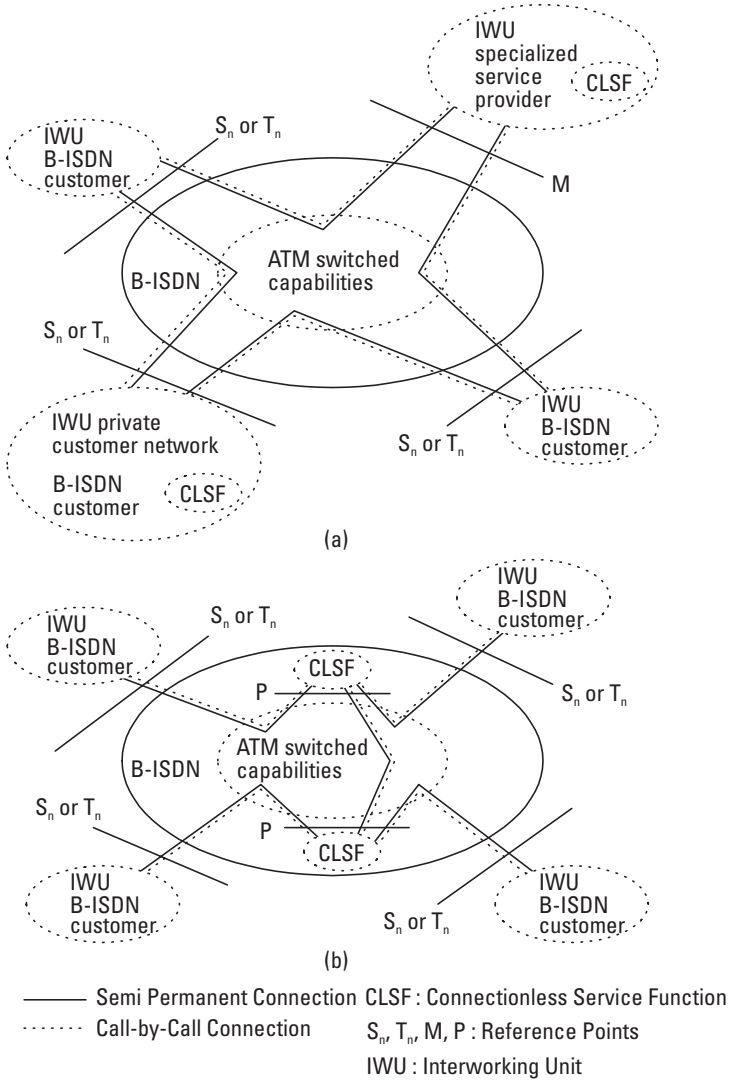

Figure 7.15 ITU-T connectionless server model: (a) indirect method, and (b) direct method.

using PVCs or may be established only when needed by using SVCs. The choice of PVC versus SVC depends on the network size and the service requirements.

SVCs can support large-size networks because it is not necessary to maintain connections when there is no data to transmit. So PVCs are used only when the size of the network is small enough to fully interconnect all IWUs with one mesh network. On the other side, SVCs require some connection setup overhead; each IWU must buffer packets until the connection is established, which causes long transmission delays. However, PVCs help to eliminate such delays.

The indirect method has the drawback (whether it uses PVCs or SVCs) that it cannot efficiently utilize the network resources, especially the bandwidth. When establishing a connection for a connectionless datagram, if the allocated bandwidth is too large, it means that bandwidth is being wasted. In contrast, if the allocated bandwidth is too small, it causes excessive delays in data transmission. Further, the indirect method makes it difficult to scale up the network since an increase of the number of end systems accompanies a rapid increase of the number of connections to support them all.

7.1.7.2 Direct Method

The direct approach can resolve the problems of scalability and bandwidth utilization. The direct method implements the CLSF using *connectionless* (CL) servers and IWUs. An IWU interconnects connectionless networks and ATM networks, and segments and reassembles the connectionless data. Each CL server may be integrated as a part of an ATM switch or may be attached to an ATM switch within the BISDN. In general, CL servers are interconnected through PVCs so that connection setup delays can be reduced. Each CL server makes routing decisions to have each connectionless datagram packet delivered to the next-hop CL server or the destination IWU.

Figure 7.16 depicts the protocol architecture of the connectionless service using CL servers. The *connectionless network access protocol* (CLNAP) in the source IWU encapsulates connectionless datagrams before delivering them to the ATM adaptation layer. The CLNAP frame is encapsulated in an AAL-3/4 convergence sublayer PDU, and is then segmented into many ATM cells. The ATM cells are delivered to the CL servers through ATM switches, and are forwarded, with or without the reassembly/processing/segmentation treatment, to the next CL server or to the destination IWU. At the NNI each CLNAP frame is encapsulated with an additional four-octet header by the *connectionless network interface protocol* (CLNIP). The *mapping entity* (ME) is responsible for the necessary encapsulation and decapsulation processes. The destination IWU reassembles the received cells into CLNAP frames and then decapsulates them into the connectionless datagrams.

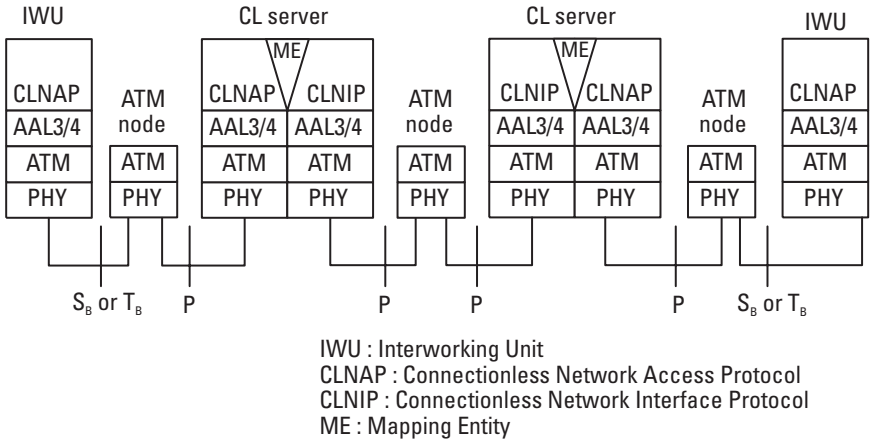

Figure 7.16 Protocol architecture of the connectionless service using CL servers.

The connectionless datagrams are finally delivered to the appropriate end systems.

The direct method has advantages over the indirect method in delivering connectionless data over public networks. First, each IWU in the direct method requires only one connection to deliver connectionless data to ATM networks, so the IWUs are not required to make the routing decisions. Second, all connectionless data traffic is aggregated in some connections between CL servers, which can increase the statistical multiplexing gain and can make the network management simple. Third, the number of required connections is much smaller in the direct method than in the indirect method because only the CL servers are interconnected, so the direct method is scalable.

On the other side, the direct method has some potential weak points: CL servers and connections between CL servers may become bottlenecks. The direct method has not resolved the complicated routing job but, instead, has shifted the job to the ATM network. There may be an interoperability problem when interoperating with LANE or IP over ATM, which has decided to use AAL-5, because the direct approach is likely to utilize AAL-3/4.

The direct method is likely to use AAL-3/4 because cell-based forwarding is much simpler than frame-based forwarding, among the two forwarding

schemes that the CL servers support. In the case of the cell-based scheme, a CL server forwards cells to the next CL server or IWU as soon as it receives them, but in the case of frame-based scheme, cells are reassembled into frames at each CL server. The reason that the cell-based forwarding scheme is required to use AAL-3/4 is as follows: When a CL server receives the first cell of a frame, it finds out the output VPI/VCI and MID in the routing table using the input VCI/VPI and MID information stored in the cell. Then the mapping information is preserved and utilized so that the other cells of the same frame, which have the same VPI/VCI and MID, can be forwarded as soon as they arrive. To keep pace with the flow of the traffic, a cell-based CL-server must be able to perform the three-phase process—receive a cell, look up the routing table, and forward the cell—within one cell transmission time. This implies that for an STM-1 155-Mbps interface all processes must be finished within 2.7 ms, and for STM-4 622-Mbps, within 680 ns.

On the other hand, in the case of the frame-based forwarding scheme, a CL server buffers all cells to be accommodated in a frame, and then reassembles them into a frame. Then it makes the routing decision or carries out other processes, and finally resegments the frame before forwarding. Since the frame is processed in one time, the MID field is not necessary, so AAL-5 can be used in the frame-based CL servers. This is why the frame-based forwarding scheme is considered to be an appropriate means for interconnecting ATM LANs. The frame-based scheme is less restricted in processing time. It can avoid useless transmissions because if a cell in the frame is lost the CL servers can detect the loss and can drop the whole frame. However, the frame-based scheme has the drawback that it requires a large reassembly buffer and causes processing delays.

7.2 Routing

The basic problem of routing can be viewed as the driving concept behind the various architectures that we have examined in the preceding section. There is a major conceptual and architectural problem in trying to use a connection-oriented network architecture to support the transport of connectionless network traffic. These methods were all advanced as a way of solving this intrinsic problem. In contrast, the problem of encapsulation (which we examine in Section 7.3) is solved in a similar manner by all the architectures. In fact they all reference the same specification, RFC1483, defined by the IETF.

There are basically two models for supporting the transport of TCP/IP traffic over ATM networks—the overlay model and the integrated (or peer-to-peer) model. LANE, MPOA, NHRP, and classical IPoA are overlay model types, and I-PNNI and MPLS are the integrated model types. The next section examines the routing issue for both models and considers how the basic problem of routing is solved in both models.

7.2.1 Overlay Model and Routing

From an architectural point of view LANE and classical IPoA are both completely aligned with the overlay models. Both use the RFC1483 encapsulation method and allow both LLC-type and VC-type multiplexing (refer to Section 7.3 for more details). This section reviews their characteristics from the perspective of how routing is performed. In addition, we examine in detail two new mechanisms recently defined to aid the autoconfiguration and operation of IPoA networks using the overlay model. One is the newly defined PAR and *proxy PAR* (PPAR) functions that offer a way for the PNNI topology database to be updated with the information on the IP routers and clients that are connected to the ATM cloud, thereby simplifying the autoconfigurability and scalability of IPoA networks. The other is the ILMI-based server discovery methods that have been defined by the IETF, which offer a way of automating the discovery of server addresses, thereby simplifying the configuration of IP clients connected to ATM networks.

7.2.1.1 Communications Within the Same Subnet

Both LANE and classical IPoA allow only ATM connections for communications between nodes in the same subnet. The nodes defining a subnet in an ATM network are named differently for each of these methods. In LANE it is called an ELAN, while in classical IPoA it is called an LIS. In both methods setting up ATM connections in the subnet requires that ATM signaling is used and that some sort of IP address-to-ATM address resolution function is available. In classical IPoA this problem was solved by the ATMARP function. As explained before, this acts as an IP address-to-ATM address resolution function. It is conceptually equivalent to the ARP function used in Ethernet subnetworks.

In LANE, the approach is slightly different as the LANE model operates at a slightly different layer than the classical IPoA model. As described before, the LANE protocol aims to appear like a LAN to the upper layers. Consequently, a node connected to a LANE will operate, at least with respect to the upper layers, as if it were connected to a LAN (either Ethernet or

token ring, depending on the configuration). This means that the node will use ARP as defined for Ethernet to find the correct MAC address to which to send any IP layer traffic. The main function in LANE is to map the MAC address of the destination node to the ATM address of the destination node.

Consequently, when a node is operating in a LANE environment the resolution of the destination address would go through a two-step process. First, the IP address of the destination would be used to find the MAC address of the destination, and then the MAC address of the destination would be used to find the ATM address. In the first step, normal ARP as defined for Ethernet/token ring would be used, and in the second step the LANE functions would be used.

For both LANE and classical IPoA, which nodes in an ATM network belong to which subnet is defined by the configuration data in the centralized servers. In LANE the servers are LES/LECS, and in classical IPoA the servers are the ATMARP servers. As the definition of subnet boundaries is dependent on the configuration data, this means that the logical boundary of a subnet does not have to be equivalent to the boundaries of the ATM network itself. This aspect brings up the concept of VLANs. In fact ELANs and LIS can both be regarded as some types of VLANs, differing only in the methods used for implementation.

7.2.1.2 Communications Between Subnets

As mentioned above, both LANE and classical IPoA only define the use of ATM connections for communicating between nodes on the same subnet (ELAN or LIS). Therefore, communicating with any node connected to some external subnet requires that a router be used as the gateway to external networks. This model essentially follows the normal routing model of TCP/IP networks. All communications inside subnets is done by methods specific to that subnet, while intersubnet routing is done through routers.

The essential problem with such a model happens when the ATM network becomes very large. In such a case it can be presumed that there are many routers connected to this network. The number of routers must be increased as the number of ATM nodes on a subnet is increased, since the router capacity is a finite resource.

If all the ATM nodes are members of a single ATM subnet, then all the routers are *adjacent* to each other. That is, they all exchange routing information with each other. The routing information will be in the form of routing protocols, such as OSPF or RIP (see Section 3.2) but this leads to a scalability problem. OSPF and RIP essentially do not work with a very large number of

adjacent neighbor routers. Having too many neighbors increases the number of protocol messages that must be sent and also increases the time for the routes to get stabilized.

To get around this problem the ATM network may be cut into reasonably small subnet groups with all communications to/from outside the subnet going through a router. However, this method brings up the possibility that even though two ATM nodes are on the same ATM network, they must communicate through a router. Note that the NHRP and MPOA protocols were defined to solve this problem, essentially by using the short-cut routing method.

7.2.1.3 PAR and PPAR

As explained in Section 4.2.2, PAR is an extension of PNNI that facilitates the distribution of information about non-ATM services in the ATM network. One example of the use for such service is overlay networks such as the ones used to support TCP/IP over ATM. Information regarding the routers, such as their IP addresses, subnet masks, and the routing protocols supported (such as RIP or OSPF) can be distributed through PAR. By using this information, it is possible for the IP routers on the edge of the ATM network to discover the other routers on the edge of the ATM network and thereby construct an IP network level topological map. This provides an efficient and dynamic way of supporting IP networks over ATM networks.

The information used in PAR is carried in a new PTSE type defined for carrying non-ATM related information. This PTSE can carry various IGs that are the actual containers of the non-ATM information. Currently IPv4-specific IGs including OSPF, BGP-4, and DNS are defined. There is also a system-capabilities IG that may be used by vendors to carry experimental or proprietary information. Table 7.2 lists the defined information groups.

PPAR. A potential problem with using PAR is that for PAR to be used in the above suggested manner the edge router must implement PNNI as well as PAR functionality. Due to the high cost of implementing PNNI, it may be better if PAR could be supported by some sort of proxy method. PPAR provides such a method. In PPAR a PAR server exists that is PAR- and PNNI- capable. Any PPAR client may then connect to this proxy PAR server and get information from it regarding the non-ATM services offered over the ATM network. The client is also able to register its own services such as its own IP address on the server. PPAR has deliberately been designed to use a separate protocol from ILMI, as the information that must be exchanged for

Table 7.2
Information Group Summary

Type	IG Name	Nested In
768	PAR service IG	PTSE (64)
776	PAR VPN ID IG	PAR service IG (768)
784	PAR IPv4 service definition IG	PAR VPN ID IG (776) / PAR service IG (768)
800	PAR IPv4 OSPF service definition IG	PAR IPv4 service definition IG (784)
801	PAR IPv4 MOSPF service definition IG	PAR IPv4 service definition IG (784)
802	PAR IPv4 BGP-4 service definition IG	PAR IPv4 service definition IG (784)
803	PAR IPv4 DNS service definition IG	PAR IPv4 service definition IG (784)
804	PAR IPv4 PIM-SM service definition IG	PAR IPv4 service definition IG (784)

PPAR operation would be much more than that which could be transferred through the use of ILMI. Of course a PPAR client may use ILMI to retrieve other configuration information.

The client/server interaction in PPAR consists of discovery, query, and registration functions. The discovery function is used by clients and servers to discover adjacent neighbors. The registration function is used by the client to register the services that it offers at the server, and the query function is used by the client to retrieve the services registered by other clients. Figure 7.17 shows an example of the operation of the interaction between a PPAR client and a PAR server. The client and server initially use the hello protocol to establish connectivity. Following this, the client registers the services that it offers (which may be the information such as its IP address) and then queries the server for information on other clients. Either side keeps the state on what information the other side may or may not have. Accordingly, the burden of maintaining correct operation falls on the client side software. The main responsibility of a PPAR server is to ensure that the information registered by the clients are delivered to all other PAR-enabled devices in the ATM cloud.

Interaction with other protocols. Neither PAR nor the PPAR specification itself defines how the service information retrieved by the client is to be used. Instead, it is expected that for each protocol, the relevant standardization organizations will define the interaction between the protocol and PAR/PPAR elements. For example, for the case of OSPF, the IETF has

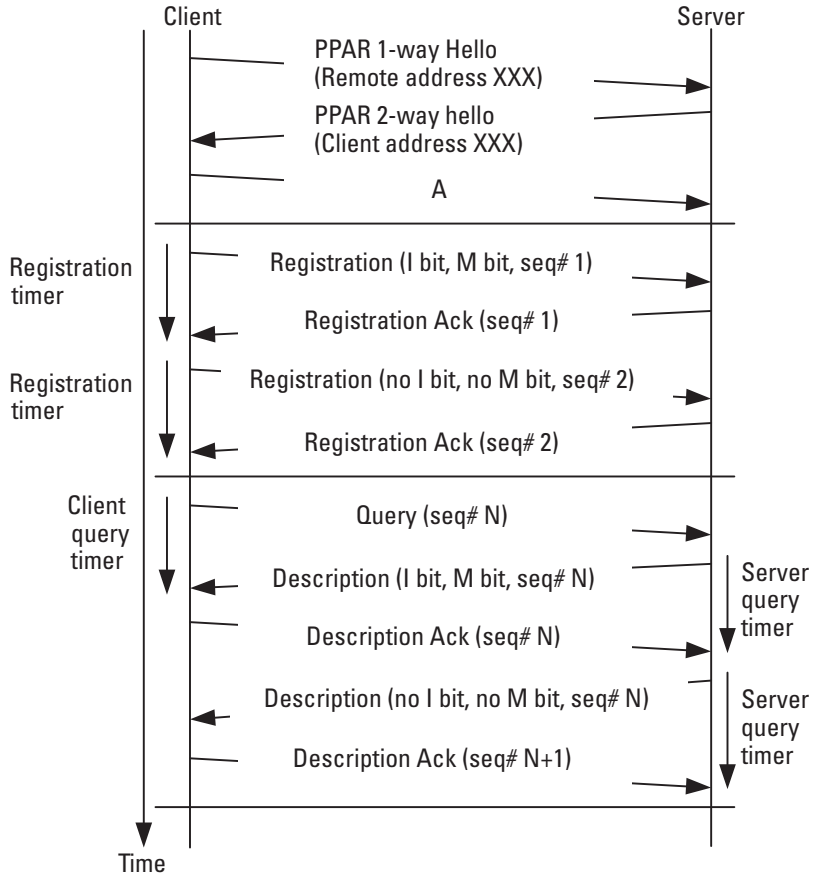

Figure 7.17 Example of the PPAR protocol operation.

defined an RFC2844 [8] that explains how OSPF is to be run over PPAR. A good example of how this may be used is the case when PPAR is used to support OSPF routing. We assume that the OSFP router is the PAR client and that a PAR server is the ATM switch to which it is connected. The OSPF IG provides information about the routers on the edge of the PNNI ATM network that supports OSPF. The information included is the OSPF area, the router priority ID, and the interface types of that OSPF router.

The interaction operation can be implemented in the following manner. For each OSPF area in which the router participates over an ATM interface, the OSPF router will transmit to the PAR server (on the ATM switch) a

PAR OSPF IPv4 service definition IG that has the OSPF and router priority information. It is assumed that the OSPF routers will use an interface type of NBMA when connected through an ATM network.[11] From the PAR server the OSPF router will get information on the other OSPF routers connected to the PNNI network. As each OSPF will have indicated its OSPF area, the OSPF routers in the same area will be informed of the presence of other routers. Based on this the OSPF routers will elect a designated router and set up VCs appropriately so as to be able to communicate with this designated router and themselves. The connection setup will be of UBR type by default unless differently configured.

Relationship with overlay models. PPAR can be viewed as a way of discovering other IP routers connected to the same ATM cloud. As explained in Section 3.2.2, a fundamental operation in IP routing protocols such as OSPF and RIP is to discover and maintain contact with neighboring routers. Usually this is achieved through the use of the hello protocol. When the neighboring routers are connected to this router by a broadcast/multicast capable networks, it is easy to design and use a hello protocol based on this broadcast/multicast ability for discovering other routers. As explained above, LANE and classical IPoA have the ability to emulate the broadcast/multicast ability by the use of servers (BUS and MARS). Consequently for both models, it is relatively straightforward to adapt the neighbor discovery functions used in the original OSPF definitions for broadcast networks.

Besides the NBMA model, OSPF also supports other network models such as the point-to-multipoint model and the simple point-to-point connections. By using PPAR it is possible to support these other network models, which usually require a large amount of configuration to be done at the OSPF enabled routers. An advantage of using PPAR in such situations is that this configuration information may be automated. For example, to use OSPF over NBMA networks the routers may be configured with the ATM addresses of all neighboring routers and with information related to the routers' OSPF settings. In the case of point-to-multipoint networks the OSPF routers are configured only with addresses of directly reachable routers. Another advantage is that the servers used in LANE on classical IPoA often represent single point of failure. By relying on PAR and PPAR the problem of single point of

11. This means that the routers will need to define a designated router to act as the representative for this network and generate LSAs to other OSPF areas.

failure disappears as the information is distributed throughout the ATM network in the PNNI database.

PAR and PPAR can also be considered to be an alternative to LANE or classical IPoA as a method for ATM address resolution. This may be an advantage as in some cases this would decrease the use of separate broadcast emulation functions defined in LANE and classical IPoA. This would result in many of the steps used to retrieve information from the ATMARP or LANE servers being skipped. To support multicast the broadcast/multicast servers are still necessary. Also, while the broadcast ability offered by LANE or classical IPoA may not be used, the information used by PAR and PPAR is always broadcast throughout the ATM cloud by being piggy-backed on top of the normal PNNI messages. Accordingly, a broadcast function must be used in some manner at some level of the protocol stack.

7.2.1.4 ILMI-Based Server Discovery

The basic classical IPoA and NHRP models both assume either that when a client first boots up the client node will have various configuration and initialization information or that it will be able to locate and connect with an initial server that can supply the needed information. The former approach is frequently subject to errors. Moreover, it is hard to reconfigure or change information dynamically as the network develops. The latter server-based approach is a more desirable method from that point of view. The ILMI-based server discovery methods defined by the IETF is such a server-based approach. The basic aim is to use ILMI as a method of automatically configuring the clients with the appropriate server addresses. Three different methods have been defined respectively for discovering ATMARP servers, NHRP servers, and MARS servers [9–11]. All these methods are similar in basic functionality, so we consider only the ATMARP server discovery method below.

ILMI offers a way for ATM attached devices to retrieve information from ATM switches and other ATM devices [9–11]. It is based on SNMPv1 and uses the *get*, *get-next*, and *trap* operations. For IP over ATM operation the ATM switch (or network side) must support service registry *management information bases* (MIBs). This is the MIB that is queried by the user (client) to get the information on the servers. The information included in the MIB includes the service identifier, full ATM address of the service, and a service parameter string. The service identifier indicates whether ATMARP, NHRP, or MARS is supported by this entry. The service parameter is service specific. For the ATMARP case, it contains information on the protocol type (IPv4 or IPv6), length of the protocol address, the network address, and the network mask. The ATMARP server must have its service registry MIBs correctly

configured with this information. It may be possible to use information supplied by the PAR protocol to keep this information updated if the ATM switch is also PAR-enabled. The ATMARP client uses the SNMP get-next operation to do a search through the service registry MIB table to find the server address needed.

7.2.2 Integrated Model and Routing

From an architectural point of view, I-PNNI and the MPLS methods are examples of the integrated model for routing TCP/IP packets over ATM networks. The integrated model is also called the peer-to-peer model as it essentially regards all the ATM nodes and IP routers as peers. They are expected to have equal and the same knowledge of the topology of the whole network, including both the ATM and IP network parts. Based on this knowledge both types of nodes are expected to set up connections or routes that will essentially take into account the whole topology. For example I-PNNI tries to be a unified routing protocol for both ATM and IP networks.

There are problems with such an approach. First of all, it is not obvious that such a method is feasible due to the differences between connectionless and connection-oriented networks. More specifically, the IP packets are routed based on the routing table. When the network topology changes, the routing table is modified automatically, changing the path between the packet. This is done by propagating the routing information and recalculating the routing tables in all the relevant routers along the path. In contrast, the paths in an ATM network will not be changed that easily since paths are set up only at connection setup time.

While both I-PNNI and MPLS may be said to be examples of the integrated model, this is not entirely correct. To be exact, I-PNNI can be viewed as an almost exact example of the integrated model, as it aims to share the topological data among separate ATM networks and IP networks, while each network essentially operates according to its own protocols. The basic operation of the ATM networks, which is based on UNI/NNI signaling, VPI/VCI table lookup, and detailed OAM, does not change. In contrast, MPLS over ATM should be viewed as a way of using ATM hardware for TCP/IP control software. It is thereby a weaker form of merger of TCP/IP and ATM. This point will become clear by recalling the operation of MPLS described in Section 7.1.6. In MPLS, the ATM protocols are essentially removed from ATM switches. ATM switches no longer use ATM UNI/NNI signaling or PNNI routing functionality in any manner, either for the setup or release of traffic flows. Instead, the ATM switch sets up its VPI/VCI tables based on the MPLS control protocols and IP routing protocols.

As we mentioned above, I-PNNI is currently not being actively pursued by the ATM Forum. In contrast, MPLS is becoming a basic solution to many problems in networking. MPLS is also being applied to solve similar problems in the optical domain (refer to Section 8.6).

7.3 Multiplexing and Switching

For IPoA networks one of the most important problems that must be addressed is the method of encapsulation: how to transport IP packets in AAL-5 payloads and how efficient is the transportation of IP traffic over ATM cells. The action of encapsulation involves adding headers and trailers to an IP packet and essentially defines how the multiplexing of the IP packets will occur in the ATM network. In the following we consider these two issues one by one.

7.3.1 Encapsulation and Multiplexing

Encapsulation of IP packets in ATM networks is defined in RFC1483. This standard defines the specific formats for using AAL-5 to transport routing protocols or the bridging of common protocols. When used to support bridging of common protocols a MAC address must be included in the encapsulation. For the cases where the aim is to only transport routing protocols no such MAC address is needed.

Basically there are two methods of encapsulation: LLC/SNAP encapsulation and VC multiplexing. In LLC/SNAP encapsulation, an LLC/SNAP header is used to multiplex different protocols over a single ATM connection. The LLC/SNAP header provides a protocol ID field, which means that even though a single ATM connection is being used, a number of different protocol connections may be multiplexed. In VC multiplexing, each protocol is carried in a separate ATM connection. In this method, it is implicitly assumed that each VC connection will carry only one packet flow. Which of the two encapsulation methods to use is decided by the configuration in the case of PVCs, and by the signaling protocols in the case of SVCs.

More specifically, LLC/SNAP encapsulation adds an LLC/SNAP header to the PDU. This was designed for the environments where VCs are scarce as in public ATM networks. It is useful in such networks because the VCs can be used more efficiently. For example, by using this method, multiple LECs can share a single VCC in the LANE architecture.

Figure 7.18(a) shows the LLC/SNAP encapsulation methods for both routed and bridged protocols. For routed protocols, the *organizational unique identifier* (OUI) in the SNAP header is set to 0x000000 to indicate

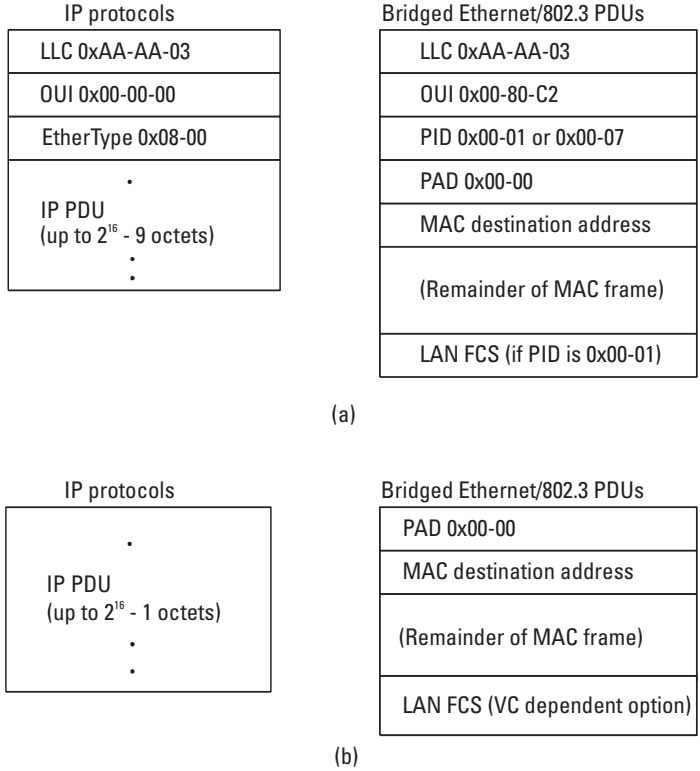

Figure 7.18 RFC1483 encapsulation methods: (a) LLC/SNAP encapsulation, and (b) VC multiplexing encapsulation.

that the PID field uses the same semantics as in the normal Ethernet [12]. For bridged protocols, the OUI is set to 0x0080C2 to indicate that the PID field follows the semantics defined in RFC1483. Two different PIDs are allowed for each case, one indicating that the LAN FCS is included, the other indicating that it is not.

As noted before, VC-based multiplexing essentially does not add any overhead to the packet itself. All users are differentiated on the basis of the VC values used. The encapsulation for such cases is shown in Figure 7.18(b), for both routed and bridged protocols. Essentially this results in supporting only a single protocol over a single ATM VC.[12] The payload efficiency is

12. Note that this does not necessarily mean that each protocol will be supported over only one platform.

high as there is no need to add an LLC/SNAP header. A disadvantage is that it requires using a large number of VCs, which will lead to scalability problems. For routed protocols, the encapsulation just consists of using the whole AAL-5 payload to transport the protocol to be routed. As noted in this section, this should be the maximally efficient method for transporting TCP/IP packets over ATM networks as there is no overhead needed. For bridged protocols, the encapsulation is similar to the LLC bridged case but without the LLC, OUI, and PID fields.

7.3.2 ATM Cell Tax Problem

The use of TCP/IP over ATM with AAL-5 encapsulation results in a large waste of bandwidth, because ATM has a fixed cell size while TCP/IP has variable length packets. The resulting wasted bandwidth is usually called the *ATM cell tax*. In general, ATM cell tax depends on the size of the original IP packet: Smaller IP packets will lose more bandwidth to ATM overhead, while larger packets will be less effected.

Normally TCP ACK packets are 40-bytes long, which consist of a 20-byte TCP header and a 20-byte IP header. In case LLC/SNAP encapsulation is used, a total of 8 bytes must be added to the TCP ACK packet. In contrast, if VC multiplexing is used, then no extra bytes need to be added. Now when this packet is encapsulated with AAL-5, an extra 8-byte overhead is incurred due to the trailers in the AAL-5 payload format. Accordingly, if LLC/SNAP encapsulation is used then the whole payload becomes 56 bytes, which is longer than what can be fit into a single ATM cell. When transported over ATM, such a payload will be divided into two ATM cells, with the second cell mostly filled with padding. In contrast, if VC-based multiplexing is used, since no extra encapsulation bytes need be added, the ACK packet may be transported in a single ATM cell. As can be seen in this example, ATM cell tax can be high when packet size is small. However, for most common IP packet sizes, for example, 576 bytes, ATM cell tax drops sufficiently low, making transport efficiency reach as high as 90% plus.

It is interesting to compare the effect of various different packet sizes. This is illustrated in Table 7.3. The table shows that while the choice of physical layer has very little effect on the efficiency, the size of the packet has a very large effect. Due to this dependency on packet size, the distribution of IP packet sizes has a large effect on the overall efficiency observed. Under the rule of thumb that 80% of the IP packets are small (e.g., 44 bytes) and 20% are large (e.g., 500 bytes), the overall efficiency becomes about 80%.

Table 7.3
Transmission Efficiency for IP Over ATM Networks

IP Datagram	Physical Layer	Efficiency	Raw Link Bandwidth	Max. Effective Bandwidth
44-byte	SONET OC-3c	39.97%	155.52 Mbps	62.16 Mbps
	SONET OC-12c	40.08%	622.08 Mbps	249.33 Mbps
	TAXI	40.00%	100 Mbps	40 Mbps
	DS-3	37.77%	44.736 Mbps	16.90 Mbps
576-byte	SONET OC-3c	80.51%	155.52 Mbps	125.21 Mbps
	SONET OC-12c	80.74%	622.08 Mbps	502.27 Mbps
	TAXI	80.56%	100 Mbps	80.56 Mbps
	DS-3	76.07%	44.736 Mbps	34.03 Mbps
1,500-byte	SONET OC-3c	85.18%	155.52 Mbps	531.38 Mbps
	SONET OC-12c	85.42%	622.08 Mbps	85.23 Mbps
	TAXI	85.23%	100 Mbps	85.23 Mbps
	DS-3	80.48%	44.736 Mbps	36.00 Mbps
9,180-byte	SONET OC-3c	86.88%	155.52 Mbps	135.12 Mbps
	SONET OC-12c	87.12%	622.08 Mbps	541.96 Mbps
	TAXI	86.93%	100 Mbps	86.93 Mbps
	DS-3	82.09%	44.736 Mbps	36.72 Mbps

The use of TCP/IP *header compression* [13] techniques is one way of mitigating the effects of the ATM cell tax. TCP/IP header compression is a method used in most TCP/IP connections running over low-speed links. It decreases the size of the TCP/IP header significantly from 40 bytes to about 8 bytes, so could be ideally suited for TCP/IP over ATM links. However, this does not necessarily justify the use of the header compression technique in the ATM links. Such a software-based header compression may not be much of a load when used over low-speed links, but becomes a big problem over high-speed links. In addition, this header compression technique assumes a different point-to-point connection for each TCP connection, such as those used over PPP or SLIP links in low-speed modem lines, but for ATM networks, this would mean that a separate VC would have to be opened up for each TCP connection. But this would cause a big problem in ATM networks as it would deplete the VCs and cause a large increase in signaling.

As far as the cell tax or link efficiency is concerned, in general, IP over SONET may be more favorable than IPoA as it also provides statistical multiplexing gain while cell tax is avoided by bypassing the ATM processing. However, it should be noted that the flexibility of the ATM, with respect to its ability to support QoS levels, a small bandwidth granularity and traffic engineering requirements through the use of VCs, is then sacrificed when using SONET links. This topic will be dealt with in more detail in Section 8.3.2.

7.4 Network Control

There are various signaling capabilities that are required for IPoA. Among them, the most notable are the capability to provide point-to-point communication with shorter on-demand setup delay, the capability to provide point-to-point communication with/without QoS guarantee, the ability to provide point-to-point communication with appropriate transfer capability and symmetric/asymmetric bandwidth, the capability to provide point- to-multipoint communication, and the ability to provide multipoint-to-multipoint communication. There are a number of problems that must be examined when investigating the interaction of ATM signaling functions with TCP/IP traffic. In this section we first examine some of these problems and then we examine the IETF specifications on using ATM signaling to set up connections to support TCP/IP traffic.

7.4.1 Basic Problems in Using VCs with TCP/IP

The basic problem in using connection-oriented VCs to support connectionless TCP/IP traffic is setting up VCs in an efficient manner. It must satisfy the network operator's need to minimize the number of active VCs at any given time. At the same time the connection setup delay should be minimized so that the delay experienced by the TCP/IP traffic and, consequently, by the user can be minimized. A number of options affect these problems with the most basic one being the question of whether to use PVCs or SVCs to support the IP traffic.

7.4.1.1 Connection Setup Delay

One of the most important requirements is the need for short ATM connection setup time. This is because ATM connection setup delay has a large effect on the performance of TCP/IP over ATM networks. It is due to the large difference between the basic conceptual architecture of the two protocol suites. TCP/IP is based on the connectionless packet network paradigm, while ATM

is a connection-oriented architecture. Connectionless packet networks do not maintain state in the network and only need to transmit packets with addresses in the headers. The network will route the packets to the correct destinations based on these addresses. However, since ATM networks always require the setup of a connection before the transmission of traffic, whenever a TCP/IP packet is to be sent over an ATM network an ATM connection must be set up beforehand.

Figure 7.19 demonstrates the effect of ATM VC setup delay for TCP/IP traffic. To send some IP data over an ATM network, the sender first sets up an ATM connection. This results in a large delay as the signaling message sent by the source has to be processed by each switch in the path to the destination. Once the destination receives the signal it then returns the signaling message. As a result, a full roundtrip time is needed to set up the ATM connection. After this connection setup data transport begins. If the user is using TCP, then another roundtrip delay is incurred for the initial TCP connection setup between the source and the destination. Consequently, the setup time of an ATM connection can substantially affect the performance of TCP/IP over ATM connections.

As an example, we consider a Web browser application. A typical Web page is made up of many different text files and image files. In HTTP1.0 every time a Web page is opened up, a TCP connection has to be set up to retrieve each text object and image object. If the connection was run over an

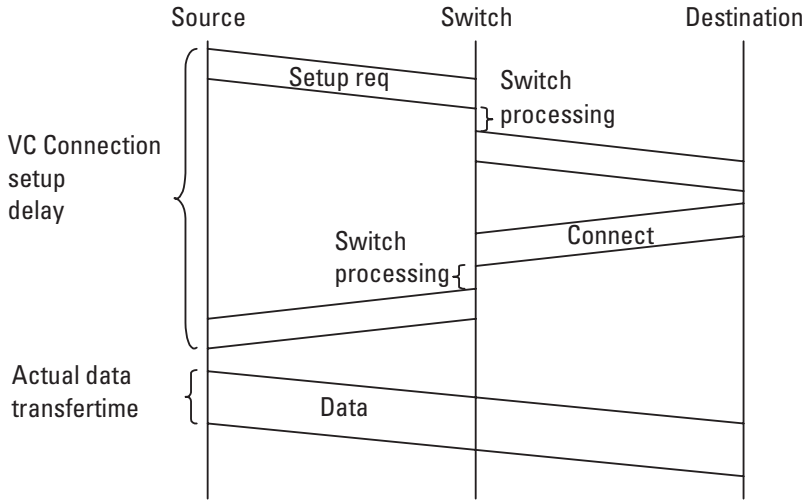

Figure 7.19 Effect of ATM VC setup delay for TCP/IP traffic.

ATM link, then in the worst case an ATM connection would have to be opened up and released for each and every text object and image object. This would increase the delay in retrieving the Web page significantly.

7.4.1.2 PVCs and SVCs

There are basically two main types of ATM virtual connections—PVCs and SVCs. PVCs are the connections that are set up and that stay up continuously. In most cases, they are set up manually by the operators. SVCs are set up on demand by using signaling mechanisms. Both types of connections have their respective advantages and disadvantages depending on a number of factors, such as the burstiness of the connection and the type of network topology in which the connections are deployed.

For bursty traffic sources, SVC may be the best fit. For such connections, the VC will probably not be needed all the time, so a PVC would be extremely inefficient. In most cases we may expect that the connection carries live data only when traffic arrives. The very definition of bursty traffic implies that for an extended period of time there may be no such traffic. In contrast, for sources such as gateways that generate steady streams of traffic, PVC may be acceptable. Though the individual traffic coming into the gateway from the various connected tributaries may be bursty, there will be a steady stream of packets that use the PVC.[13]

Using PVCs. If PVCs are used, the ATM connection setup time normally has no effect as the connection is a permanent one and will have no discernible effect on the total time observed by the user. This will be especially true for cases in which manually configured static PVCs are used. If soft PVCs are used, then some signaling will occur, but it will only occur when the network is in an unstable state or a link is down. That is, these events will be unsynchronized with the normal request for connection sent out by normal calls. As such, they can be expected to have no effect on the overall performance perceived by the user.

The use of PVCs in either of the above forms is suitable for small networks. Once the network becomes large the use of PVCs to interconnect the sites becomes very hard to manage. Basically, such a solution has problems in scaling to very large networks. The operator of the network would have to

13. This does not necessarily imply that the stream will be steady in CBR. Mixing bursty traffic may often result in more bursty traffic. This phenomenon is studied in the field of "self-similar" traffic research.

configure the connections on each switch and router. Whenever a new router or edge device is added the PVCs to all the other devices would also have to be reset up to ensure routability. This leads to the classical N^2 complexity problem.

Using SVCs. If SVCs are used, it brings in a signaling load every time an SVC is set up. Accordingly, the frequency of SVC setup will be an important determining factor of the overall performance; how often SVCs will be set up and how fast SVCs can be set up are the major factors. How fast the network sets up SVC connection is an implementation issue that could vary depending on switch manufacturers. As such, we cannot pursue this aspect of the problem any further. If the switches are capable of handling a large amount of signaling load, it is not necessary to attempt to minimize the signaling load.

The important question here is how to decide when to set up an SVC. This determines the frequency of SVC setup and consequently the performance perceived by the end user. This decision may be done in a number of ways. The simplest way is to allow the first packet to always trigger a connection setup. Another way is to use the amount of packets transmitted as a trigger. Also available are more complicated methods that depend on the current state of the network and router.

Another related issue is when the connection should be torn down. This is really the other side of the previous problem. In most cases, this will also rely on a timer, or an inactivity timer. The inactivity timer is used in such a way that if no traffic is observed over the connection for a certain amount of time the connection is released. Any time the connection carries real data the connection is reset. So the timer value has a critical effect on the performance and the efficiency of the connection. If the timer is too long, the connection will be kept alive even though there is no traffic to transmit, resulting in a waste of resources in the ATM switches along the path. If the timer is too short, too much signaling processing will take place for the setup and release of the ATM connections.

There are two basic approaches that can be obtained from the above methods. One is to set up SVC when there are packets to transmit but to delay the closing of the connection until the inactivity timer goes off. The other is to delay opening the connection until a certain amount of packets has arrived and been buffered but to close the connection immediately after all the currently buffered packets have been sent. The former minimizes the delay that the first packet will experience, but at the cost of network efficiency. The latter ensures that the opened connection is used efficiently but sacrifices delay. There are various other approaches that have been made by

using queuing theory and simulation tools, but the results vary depending on the applied assumptions.

7.4.2 Signaling Support for Classical IPoA and NHRP

The signaling support defined by the IETF in RFC1755 defines the signaling methods that must be used to set up connections to transport best-effort traffic, while the signaling support defined in RFC2381 (and the related specifications RFC2379, RFC2380, and RFC2382) defines the same information for controlled-load service and guaranteed service. Note that these specifications essentially define how the IntServ model of IP QoS is to be supported when TCP/IP is used over an ATM network.

RFC1755 and RFC2381 describe the ATM call control signaling exchanges needed to support classical IPoA implementations as described in RFC 1577 [1]. ATM endpoints are assumed to incorporate the ATM signaling services as specified in the ATM Forum's *UNI Specification Version 3.1 and 4.0* [14, 15]. Clients such as classical IPoA, LANE, and MPOA implementations utilize the services of local ATM signaling entities to establish and release ATM connections to support IP traffic.

7.4.2.1 VC Establishment

The owner of an existing VCC is defined to be the entity within the ATM end system that establishes the connection. An ATM end system may establish an ATM call when it has a datagram to send but there is no existing VCC that it can use for this purpose or the VCC owner does not allow sharing.

To reduce the latency of the address resolution procedure at the called station, the following procedure may be used: If a VCC is established using the LLC/SNAP encapsulation, the calling end station of the VCC may send an InARP_REQUEST to the called end station after the connection is established (i.e., received a CONNECT message) but before the calling end station sends the first data packet. In addition, the calling end station may send its data packets without waiting for the InARP_REPLY. An end station may respond, generate, and manage its ATMARP table according to the procedures specified in RFC1293 [2], during the lifetime of the VCC.

To avoid establishing multiple VCCs to the same end station, a called end station may associate the calling party number in the SETUP message with the established VCC. This VCC may be used to transmit data packets destined to an end station whose ATMARP resolution results in an ATM address that is the same as the associated calling party number.

Support for multiple VCs. An ATMARP server or client may establish an ATM call when it has a datagram to send but there is no existing VCC that it can use for this purpose, it chooses not to use an existing VCC, or the owner of the VCC does not allow sharing. Note that there might be VCCs to the destination that are used for IP, but an ARP server might prefer to use a separate VCC for ARP only. The ATMARP server or client may maintain or release the call as specified in RFC 1577. However, if the VCC is shared among several protocol entities, the ATMARP client or server does not disconnect the call as suggested in RFC 1577.

While allowing multiple connections is specifically desired and allowed, implementations may choose (by configuration) to permit only a single connection to some destinations. In such a case, if a colliding incoming call is received while a call request is pending, the incoming call is rejected. Note that this may result in a failure to establish a connection. In such a case, it is recommended that each system wait at least a configurable collision retry time in the range of 1 to 10 seconds before retrying.

7.4.2.2 VC Teardown

Either end system may close the ATM connection. Systems configure a minimum holding time (i.e., the time the connection has been open) for connections to remain open as long as the endpoints are up. A suggested default value for the minimum holding time is 60 seconds.

Some public networks may charge for connection holding time, and connections may be a scarce resource in some networks or end systems. Accordingly, each system implementing a public ATM UNI interface should support the use of a configurable inactivity timer to clear connections that are idle for some period of time. The timer's range includes a range from a small number of minutes to "infinite." A default value of 20 minutes is recommended in RFC1755.

7.4.2.3 Call Establishment Message Content

Signaling messages contain mandatory and optional variable-length *information elements* (IEs). The IEs are further subdivided into octet groups, which in turn are divided into fields. IEs contain information related to the call, which is relevant to the network, the peer end point, or both. The called end station and the type of communication channel opened over the ATM connection are determined by the IEs that are contained in the call establishment message. For example, the call establishment messages will differ between a call that sets up an AAL-1 connection for CBR video and a call that sets up an AAL-5 connection for IP.

A SETUP message that establishes an ATM connection to be used for IP and multiprotocol interconnection calls must contain the following IEs: AAL parameters, an ATM traffic descriptor, broadband bearer capability, broadband low-layer information, a QoS parameter, the called party number, and the calling party number.

There are IEs in a SETUP message which are important only to the endpoints of an ATM call supporting IP. These are the AAL parameter IE and the broadband low-layer information IE; the AAL parameter IE carries information about the AAL to be used on the connection. RFC 1483 specifies encapsulation of IP over AAL-5. Selection of an encapsulation to support IP over an ATM VCC is done by using the *broadband low layer information* (B-LLI) IE.

7.4.2.4 ATM Traffic Descriptor

The ATM traffic descriptor characterizes the ATM virtual connection in terms of PCR, SCR, and maximum burst size. This information is used to allocate resources (e.g., bandwidth and buffer) in the network. In general, the ATM traffic descriptor for supporting multiprotocol interconnection over ATM will be defined based on factors such as the capacity of the network, conformance definition supported by the network, performance of the ATM end system, and (for public networks) cost of services.

The default model of IP behavior corresponds to the best-effort capability. If this capability is offered by the ATM networks, it may be requested by including the best-effort indicator, the PCR-forward, and PCR-backward fields in the ATM traffic descriptor IE. When the best-effort capability is used, the network does not provide any guarantees, and in fact, throughput may be zero at any time. This type of behavior is also described by RFC 1633 [16].

If the user (or network) desires to use a more predictable ATM service for IP traffic, it must be possible to use more specific traffic parameters. In such cases the basic traffic descriptor IE is used along with a broadband bearer capability IE and the QoS parameter IE. These elements essentially define the signaling aspects of ATM traffic management.[14]

14. The specific elements are defined in a number of RFCs including RFC1755, RFC2331, and RFC2381. RFC1755 also defines a set of combinations of traffic parameters that the ATM signaling modules in all IPoA end systems must support. These include the best-effort traffic, a type of traffic description that is intended for ATM "pipes" between two routers or IP systems, and a type of traffic description that allows use of token-bucket style characterizations of the RFC1363 and RFC1633.

7.5 Traffic Management

When IP is used over an ATM network various traffic problems appear that degrade the network efficiency. They basically stem from the inherent differences between IP and ATM—connectionless against connection-oriented, fixed-length against variable-length, and so on. To resolve such traffic problems and thereby enhance the network efficiency, it is important to examine the effects of ATM cell loss on TCP/IP performance and then devise methods to handle the resulting limitations in proper ways.

7.5.1 Effects of ATM Cell Loss on TCP/IP Performance

In a seminal study, it was shown that if ATM networks are used to transport TCP traffic without any special provisions, then the throughput seen by the TCP/IP user could become very low, in some cases dropping to 34% of the available link capacity [17]. This effect is basically due to the fact that the ATM layer has no knowledge of the packets carried by the upper layers. This can be analyzed in more detail as follows.

A single TCP segment is usually transported in one IP packet, whose normal size depends on the path MTU but is usually 1,500 bytes or 576 bytes. As this single TCP segment does not fit into a single ATM cell in general, it is segmented into plural ATM cells after going through an appropriate encapsulation process. (Refer to Section 7.3 for more details on encapsulating TCP packets in ATM cells.) The cells are then transported across the network and, when all the cells for a packet are received by the destination, they are reassembled into the original packet. The destination recognizes the last cell of a packet by examining the packet indicator in the ATM cell header. Once the last cell is determined, the reassembler engine assumes that all the cells have arrived, checks the 32-bit CRC field at the end and, if it is correct, delivers the packet to the upper layer application.

When the ATM cells happen to confront network congestion while traversing the ATM network, some cells may be dropped. If this happens, the reassembler in the receiver will find that the CRC value does not match, and the packet will then be dropped. This is the only mechanism that can take place at the node as ATM does not offer any retransmission or upper-layer error recovery mechanism. Consequently, even a single ATM cell loss can result in dropping of the whole packet.

Such a large multiplicative effect decreases the efficiency of the network operation. First, it means that all the ATM cells belonging to the dropped packet are discarded even if they were successfully delivered.

Second, it means that the packet must be retransmitted, with a high probability of confronting congestion again and thereby suffering cell loss. More importantly, it triggers TCP's congestion control mechanism, pushing TCP connections into slow-start mode. (Refer to Section 3.5 for the basic TCP congestion control mechanism.)

Various solutions to this problem have been suggested. Simplest among them may be to increase the buffer sizes in the network to minimize the cell loss due to congestion. However, this solution is not very efficient and possibly leads to a large network delay, which is not conducive to effective network operation. Basically, there are two approaches that have been proposed to deal with this problem. One is to use a feedback mechanism to ensure that the cell loss due to congestion is minimized, which is the basis of the ABR rate control mechanism examined in Section 4.5.2. We examine this approach in Section 7.5.2. The other approach is to use a frame discard mechanism to ensure that unnecessary cell transmissions do not take place. We consider this approach in the following.[15]

7.5.1.1 Frame Discard Methods

There are two different types of frame discard, one is *partial packet discard* (PPD) and the other is *early packet discard* (EPD). Both PPD and EPD basically try to emulate the loss behavior observed in normal packet networks and thereby minimize the bandwidth used by the traffic that is useless at destination. The main difference is in the decision principle on which cells to drop once congestion has been detected.

The PPD method is to immediately drop the cell and all the following cells with the same VPI/VCI until a cell with an end-of-packet indication is found. This method will ensure that no useless cells are transmitted through the congested switch beyond those already sent before the congestion was detected. Note that to ensure successful operation the cells with the end-of-packet indication bit must not be dropped and reach the destination. Otherwise, the next packet would also be dropped due to CRC check failure.

The EPD method is to drop a whole packet, or, to be more precise, all the cells that carry the segmented data of the next whole packet if the queue size grows larger than the EPD threshold. This mechanism is illustrated in

15. Note that all the cells carrying the data belonging to a single packet must have the same VPI/VCI, as they must all be transported over the same virtual circuit.

Figure 7.20. This operation can be easily accomplished by looking at the incoming cells for the next end-of-packet indication and then dropping the cells incoming from that point on until the next end-of-packet indication.

Table 7.4 compares the performance of the PPD and EPD methods along with the normal packet network TCP [18]. The normal UBR case is denoted as UBR-PLAIN, the implementation using PPD as UBR-PPD, and the implementation using EPD as UBR-EPD. The last is the case when a normal packet network TCP is used. We observe that the throughput increases steadily from the UBR-PLAIN to the UBR-PPD, to the UBR-EPD, and then to the packet TCP. In the end, the UBR-EPD and the packet TCP cases basically exhibit the same throughput.

Two of the critical issues in the frame discard methods are the criteria to decide when to drop the cell of a frame, and which VPI/VCI cells to drop. The first issue is usually solved in a straightforward manner by using a threshold value and comparing the current queue length in the switch.

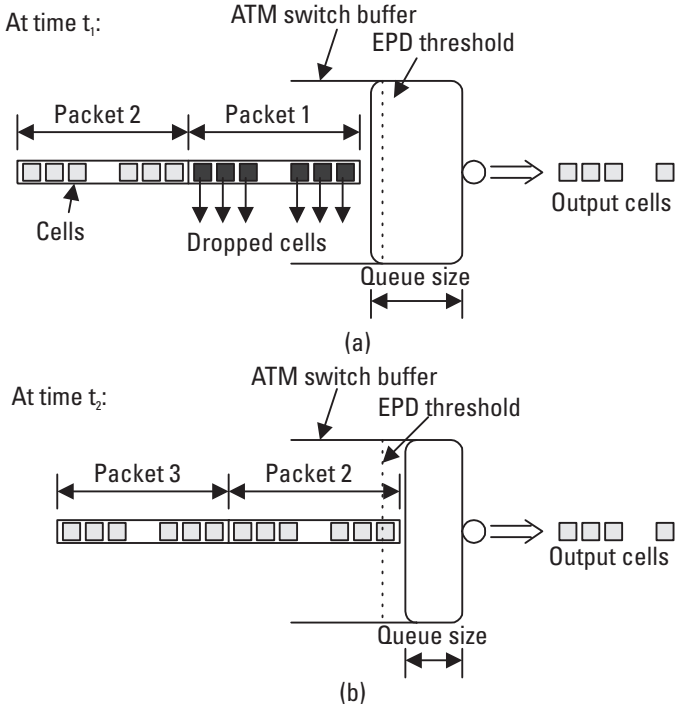

Figure 7.20 Operation of EPD algorithm: (a) when the queue size is larger than the EPD threshold, and (b) when the queue size is smaller than the EPD threshold.

Table 7.4
TCP Throughput over Various Flavors of UBR Service

Buffer Size (in Kbytes)	UBR-PLAIN	UBR-PPD	UBR-EPD	Packet TCP
100	0.62	0.8	0.98	0.98
200	0.7	0.88	0.98	0.98
300	0.8	0.92	0.98	0.98
400	0.84	0.95	0.98	0.98

From: [18].

However, the second issue is rather tricky, as it is related to the fairness in packet dropping. As noted in Section 3.5.2, the queue management policies will have a large effect on the end throughput seen by each user and consequently the fairness of the whole system.

7.5.1.2 GFR-Based Methods

GFR is a new traffic management category defined by the ATM Forum to more efficiently support TCP/IP traffic-based on the initial research on frame discard methods and other optimizations [19]. GFR originates from the idea of making the frame discard mandatory and giving to TCP/IP traffic a guarantee of the minimum rate as well. By making frame discard mandatory for all switches, GFR ensures user level performance to be better than pure UBR switches. Moreover, GFR allows the user to reserve some bandwidth in the network. This value usually equals the minimum rate that the relevant application would need.

The GFR traffic contract contains PCR, MCR, MBS, and MFS attributes.[16] An important characteristic of GFR is that the end system is not required to shape the traffic. Consequently a packet will usually be transmitted as a burst of cells at the PCR. For policing such a traffic pattern the normal GCRA must be modified. *The frame-based GCRA* (F-GCRA) was defined for this purpose [19]. While the ideal F-GCRA algorithm is exact, it is not very practical, so the simple F-GCRA(T, L) algorithm was also

16. The MFS is the size of the AAL-5 that is used. The MCR is usually negotiated to be equal to the long-term average of the connection.

introduced in the specifications. Figure 7.21 describes this algorithm. While it is a simplified form of the ideal F-GCRA algorithm, it is equivalent to the ideal F-GCRA(T, L) algorithm for the connections that only contain conforming frames or cells.

GFR has a number of advantages over other service classes in supporting TCP/IP traffic. Compared with UBR, GFR is more efficient as the network supports frame-discard strategies. Compared to VBR, GFR has advantages as QoS is guaranteed at the frame level, which is a more meaningful metric for the end user. In contrast to ABR, GFR imposes no complex scheduling or queuing mechanisms on the end points. Nor does GFR need to use RM cells as ABR does.

7.5.2 Using ABR for TCP/IP Traffic

The feedback mechanism renders a viable solution to traffic control when transporting TCP/IP traffic over ATM network. TCP/IP traffic is bursty and not ideal for rate-guaranteed traffic contracts and efficient transport over virtual circuits. In contrast, ATM is optimized for transporting fixed rated traffic over configured virtual circuits. The feedback-based traffic control mechanisms used in this situation help to enhance network efficiency by preventing network congestion by controlling the data rate and thereby avoiding ATM cell dropping.

The ATM Forum has defined a new parameter, *minimum desired cell rate* (MDCR), to use during call setup to indicate to the network that a TCP/IP application would like to get a certain minimum bandwidth [20].

Cell arrival at time t_a:	
First cell of an AAL-5 frame:	Middle or last cell of an AAL-5 frame:

if((t_a < *TAT*- *L*) OR (IsCLP(cell)) {
/* non-eligible cell */
eligible = FALSE;
} else {
/* eligible cell */
eligible = TRUE;
TAT = *max*(t_a, *TAT*)+*T*;
}
	if(eligible) {
	/* eligible cell */
	TAT = *max*(t_a, *TAT*)+*T*;
	} else {
	/* non-eligible cell */
	}

Figure 7.21 Simple F-GCRA (T, L) algorithm.

The MDCR differs from other ATM traffic parameters in that it does not define a service commitment to guarantee the minimal requested rate. For example, if a user requests a connection with a certain MDCR value, the network internally commits a minimum bandwidth of MDCR to the connection, and polices the traffic according to the PCR with a GCRA (1/MDCR, T) test.

In contrast to the MDCR-based rate control, ABR-based rate control is an active form of feedback-based traffic control mechanism. It defines a structure by which data traffic can fairly and efficiently share all the resources of a network, by controlling the data rate based on the feedback information. Accordingly, it is natural to expect much enhanced network efficiency by ABR rate-controlled VCs to support TCP/IP traffic. However, the use of ABR rate control brings in a number of problems yet to be solved as well. In the following section we will discuss the ABR-based traffic control in detail.

7.5.2.1 Effects of ABR Rate Control

When ABR rate control was first proposed, extensive simulations were carried out to show that the use of ABR rate control results in an efficient transport of TCP/IP traffic over ATM networks. Many results indeed exhibited the network utilization sustaining nearly 100% without any cell loss or abnormal delay. However, a key point to note on the simulations is that an end-to-end ATM network was assumed. In reality, however, ATM networks do not appear at the desktop but are mostly used to interconnect various LAN-based IP edge networks; this is therefore called the *subnet ATM model.* This means that the ATM connection employing ABR flow control is normally used between LAN ATM edge devices. This introduces an interesting interaction phenomenon between TCP's congestion control and ABR rate control: Even if, as noted before, the use of ABR in an all-ATM network may ensure a maximum network utilization and fair network resources sharing, neither efficiency nor fairness can be guaranteed when the ATM network, or the consequent ABR feedback loop, covers only part the network. In the following we shall call such a basic interconnection scheme for transporting TCP/IP traffic over ABR connections the *ABR+ scheme*.

Figure 7.22 shows an ATM network implementing ABR rate control to connect to IP LAN networks. If congestion occurs in the ATM network, then the ABR rate control mechanism signals the edge LAN ATM devices that congestion has developed in the network so the rate of transmission should be reduced. This should effectively cause the congestion in the ATM

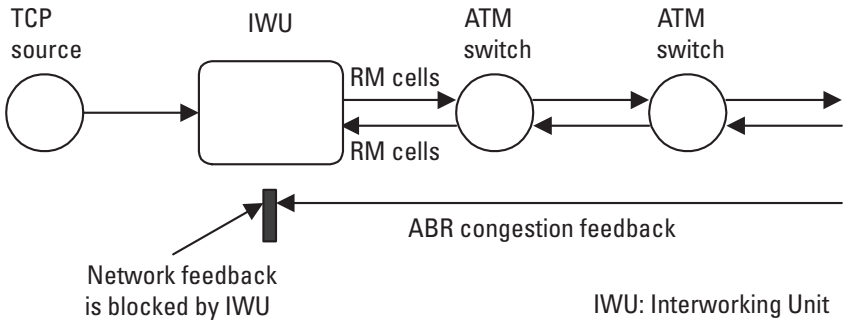

Figure 7.22 Effect of using ABR connections to support TCP/IP traffic.

network to disappear. However, what really happens is that congestion then starts building up in the LAN ATM device. In other words, the use of ABR simply shifts the point of congestion from inside the ATM network to the edge of the ATM network.

When all nodes are ATM capable, then "congestion" occurs inside the source node. This is not a problem, as it will only result in the source sending out cells/packets more slowly, with no cell or packet loss occurring at the source. The real problem takes place when the congestion occurs at the LAN ATM device, with the IP source being a completely separate device. In this case, the IP source does not receive any information that congestion has occurred, so does not slow down the transmission rate. In fact, if it is a TCP implementation, it will slowly increase the rate, only aggravating the situation and resulting in more packet losses at the LAN ATM device. In the end, TCP implementation will know congestion has occurred, but only after the congestion point has moved from the inside of the network to the LAN ATM device. As the buffers at the LAN ATM device can be expected to be large, this will cause a very large feedback delay even before the source TCP realizes congestion has occurred.

There is another facet to this problem that has to do with the fairness of the bandwidth usage. Since one of the basic aims of the ABR mechanism was to share the available bandwidth fairly among the connections, fair allocation of bandwidth should be easily attained. Also, as ABR aims to maintain a minimal or zero cell loss rate, the fragmentation effect should not be a problem. In the end-to-end connection case, it has been shown that almost perfect fairness and high throughput can be achieved. However, the same result does not automatically carry over into the subnet ATM case. This is because

the gateway at the edge of the ATM cloud acts as a barrier to the feedback coming from the network, consequently nullifying the ABR mechanism by hiding the network congestion signals coming from the end host.

ABR expects the traffic source to control its output rate according to the feedback information. However, in the subnet ATM model, though the gateway will immediately drop its output rate according to the ABR feedback messages, this will only cause a queue buildup at the gateway. The TCP source will lower its output rate only after packet loss occurs in the gateway due to the overflow of this queue. Unless some special buffer management schemes are employed at the gateway, the connection whose packets are dropped will not necessarily be the connection whose rate was lowered by the ABR mechanism. This leads to the wrong connection decreasing its window size and thereby its output rate. The effects of this behavior become predominant when large and small delay connections share a single output buffer. In this case the ABR+ scheme behaves unfairly and with low goodput.

Basically, this problem is due to *hogging* of the available buffer space by the short delay connections and the dynamic allocation of service rates of the ABR mechanism. The short delay connections will fill up the available buffer space, causing the long delay connections to be squeezed out of the buffer and leaving an uneven distribution of packets in the buffers. However, as the link bandwidth is fairly divided among the connections by the ABR mechanism, each connection will be served at its allocated rate. Consequently, there is a shortage of packets when the long delay connections queues are served, and hence the ABR+ scheme becomes underutilized and unfair. This leads to the allocated bandwidth not being fully used, since the long delay connections see less buffer space than they need to fully utilize their allocated bandwidth. This effect is especially predominant with relatively small buffers since the short delay connections can easily fill up the small amount of available buffer space.

In summary, though ABR rate control may *allocate* the available bandwidth fairly among the connections, it does not guarantee that the connections will be able to effectively *use* the allocated bandwidth. This is basically due to a mismatch between the rate control-based ABR mechanism, which relies on explicit rate feedback, and the window-based TCP congestion control mechanism, which relies on implicit feedback in the form of packet loss.

The fact that ABR rate control pushes the queues to the edges of the network may also be helpful in solving the problem it creates. Since the gateway now becomes the single point of congestion for that connection, we can concentrate on designing an intelligent gateway congestion control scheme that utilizes the ABR rate feedback information. The basic idea is to design a

fair buffer allocation scheme so that the bandwidth allocated by the ABR mechanism may be fully utilized.

7.5.2.2 ABR with Fair Buffer Allocation (ABR+FB)

A number of solutions have been proposed to solve this problem. Two of the main solutions are based on the observation that to improve efficiency some sort of intelligent queue control algorithm must be implemented in the LAN ATM device.

Jagannath and Yin proposed such an intelligent approach [21]. In their approach they do not wait until the queue is full, but drop a packet immediately after the ABR rate control signals a rate decrease. Additionally, this drop is implemented by using a drop from front policy, thereby minimizing the delay in notification to the TCP source. It has been shown that this method is effective.

Kim and Lee proposed a more complicated but more efficient scheme by noting that for an effective ABR rate control some method must be devised to reflect the rate information in the congestion control scheme at the gateway on the edge of the ATM cloud [22]. They proposed a *fair buffer* (FB) allocation scheme that aims to use this feedback information and obtain better utilization of the allocated bandwidth. Along with the fair bandwidth allocation due to ABR, this resulted in fair and high throughput for the TCP connections. This method is called the *ABR+FB scheme*.

The fair buffering mechanism is based on the following two observations. First, the total achieved throughput of a TCP connection with a fixed service rate depends only on the normalized buffer size, and is maximized when its buffer space exactly equals the connection's bandwidth-delay product.[17] Consequently, for fair utilization of the allocated bandwidth by each connection, buffer space must be allocated in proportion to its *bandwidth-delay product.* Second, the TCP congestion control mechanism works most effectively when its loss is spread out and does not occur in bursts. The question of how much the loss should be spread out can be decided by observing that TCP Reno operates optimally when there is only one loss per congestion

17. The ABR+FB algorithm relies on the correct calculation of the bandwidth-delay product. In traditional best-effort networks, calculating the bandwidth-delay product for a connection was not possible as neither the bandwidth nor the delay of the connection was known. In contrast, when ABR connections are used, though bandwidth is variable over time, it is explicitly and fairly allocated for each connection during discrete intervals. Also, as ATM is connection-oriented, the fixed propagation delay of the path is also known.

avoidance cycle. Ideally, if one packet per congestion avoidance cycle is dropped, this should be optimal.

The ABR+FB scheme is based on the following observations. First, to filter out noise due to rate fluctuations, it is desirable to use the average of the bandwidth in all calculations. Second, it is necessary to allocate buffer space in proportion to the bandwidth-delay product. Third, it is important to minimize the problems that occur during the slow-start congestion avoidance cycle when the connection is started. However, the ABR+FB scheme relies on the characteristics of TCP congestion control, so may not effectively control UDP-based traffic.

7.6 QoS

As discussed in Section 3.6, there are two different models defined by the IETF for QoS support in TCP/IP networks: IntServ and DiffServ. The IntServ QoS model is logically very similar to the ATM model in that it relies on specific resource allocation, resource reservation, and a signaling protocol to guarantee services to users. As such, the model offers a relatively obvious logical model of integration, whereby the resource allocation and reservation information must be mapped to appropriate ATM functions, most importantly, the signaling aspects. An important condition for supporting the IntServ model is that the RSVP path messages must follow the same path as the path used by the data traffic towards the destination. The DiffServ QoS model is slightly different in that it relies on marking individual packets to be classified into aggregate classes. It is a more incremental approach that is more compatible with the current state of TCP/IP networks and technology.

Additionally, there are two basic models for supporting IP traffic over ATM networks: the overlay models and the integrated model. While there are a number of methods classified as being based on the overlay model, there is only one existing method based on the integrated model, that of MPLS over ATM. As the integrated model aims to integrate the ATM switches and IP routers on a peer-to-peer basis, this means that the same control function must be used on all nodes. This means that the QoS functionality must be supported in a similar manner on both ATM and IP nodes. In other words, all nodes must be aware of the MPLS control traffic, some of the IP control traffic, and the QoS classes and services supported in the network. In contrast, in the overlay model, while the IP routers on the edge of the ATM network must be aware of such information, the ATM switches internal to the network may not know of the IP packet's QoS class. Instead, such

information must be passed in an indirect manner, which usually consists of setting up appropriate connections based on a mapping of IntServ parameters to ATM QoS parameters.

Various mixes of models for supporting QoS and IPoA are shown in Figure 7.23. The mapping of the two QoS models with the two IPoA models results in four different basic models. Each model uses a slightly different solution to the basic problem. The overlay models map the QoS of IP connections to ATM connection type, while the integrated model relies on the ATM switches understanding and processing RSVP messages or carrying IP level information (e.g., DSCP values) in the ATM cells themselves. We discuss Figure 7.23 in more detail in the following sections, where we examine the specific solutions.

7.6.1 Overlay Model and QoS Support

As mentioned above, from a QoS point of view the overlay model basically uses the ATM protocol suite as currently defined, employing ATM connections as layer 2 links over which IP packets are transmitted. In such a model the routers on the edge of the ATM cloud, or the edge routers, must carry out the processing needed to support QoS interaction between the ATM network and the IP network.

QoS models / IPoA models	IntServ	DiffServ
Overlay	Mapping between ATM connection & RSVP info.	Differentiated UBR
Integrated	Carry RSVP information	Map DSCP to ATM cell header

Figure 7.23 Mapping IPoA models to QoS models.

Figure 7.24 shows the edge router functions in an abstract manner. The edge router must be able to maintain multiple VCs per IP flow and maintain mappings of QoS classes to the appropriate VCs, and implement sophisticated queuing and scheduling algorithms. In addition, if IntServ is supported, it must be able to translate QoS parameters between those used by RSVP and those used in ATM. The edge router must be able to use the information contained in the RSVP signaling message to guide the establishment of ATM VCs of the appropriate QoS class toward the next-hop router connected to the same ATM cloud. If DiffServ is to be supported, the edge router must be able to map the DSCP and PHB of the flow to the appropriate output queues and scheduling classes.

Figure 7.25 shows on a larger scale how the edger router functions and the various QoS techniques developed for ATM can be used in a large IP over ATM network to achieve the aims of supporting QoS for IP traffic. The key is to have most of the complicated QoS functionality in the edge routers with minimal functionality (mostly congestion control functions) kept in the core of the ATM network itself.

A number of problems must be addressed in this model. One problem is the management and use of VCs, which was dealt with in Section 7.4. A key question that must be answered is how the mapping between the IP QoS service classes and the relevant ATM QoS classes should be done. Table 7.5 shows how the mapping between ATM QoS classes and the various classes in the IntServ and the DiffServ models can be done. It shows that ATM QoS classes defined in UNI 4.0 and TM 4.1 can be used to support both types of IP QoS models. In addition, the new differentiated UBR service may be used

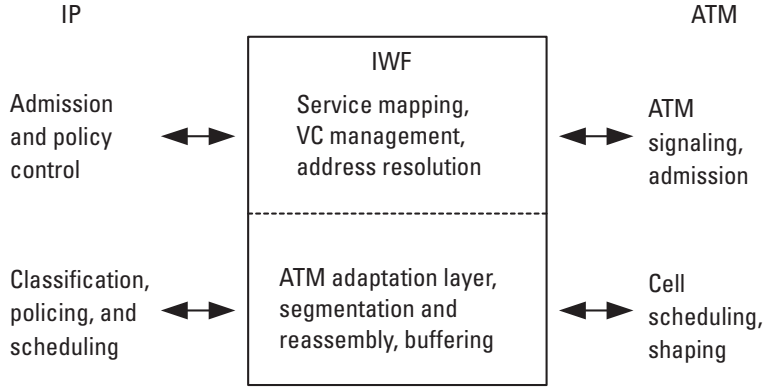

Figure 7.24 Edge router functions [23].

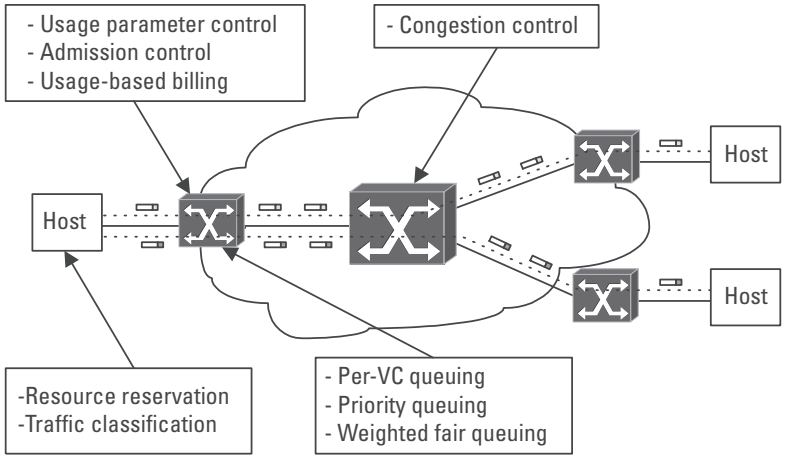

Figure 7.25 Example of different techniques used to support QoS in ATM networks.

to support DiffServ QoS in a more natural manner [24]. However, Table 7.5 does not necessarily provide an absolute guide, as there may be a number of ways of mapping the ATM QoS classes to the IP QoS classes.

7.6.1.1 Overlay Model and IntServ over ATM (UNI 4.0)

To use the IntServ QoS model over networks using the overlay IP over ATM model a number of problems must be resolved, including VC management and QoS mapping. The critical condition of the RSVP path messages that follow the same path as the forward data path is satisfied by ensuring that the ingress and egress points of the path over the ATM network is the same for the RSVP messages as well as the data packets. Note that the actual path

Table 7.5
Different Ways of Mapping IP QoS with ATM QoS in the Overlay Model

QoS Model	Service Class	ATM QoS Class
IntServ	Guaranteed service	CBR or rt-VBR
	Controlled-load service	nrt-VBR or ABR (MCR)
	Best-effort service	UBR or ABR
DiffServ	Expedited forwarding (EF)	CBR or rt-VBR or UBR (diff)
	Assured forwarding (AF)	nrt-VBR or ABR or UBR (diff)

taken inside the ATM path need not be the same for the two types of packets, as the information needed to set up QoS paths for the data packets is only needed at the ingress and egress points. The default best-effort VC will always be established before any QoS-specific VCs as it is needed to transport the initial RSVP PATH messages to the receivers. The initial RSVP RESV messages are also received along this VC.

One of the main differences between the ATM QoS and IntServ is the signaling model. RSVP is a receiver-oriented protocol where the receiver signals to the routers along the path what amount of resources must be reserved, while in ATM it is the sender that decides the amount of resources to reserve. Actually, this is not a big difference because the RSVP receiver bases its reservation decision on the sender's TSPEC that was included in the PATH message. Consequently, the reservation may also be viewed as being initiated by the sender in an indirect manner.

Another important problem is how to manage the VCs that must be set up to support different QoS levels and connection topologies. As discussed in Section 7.4, two types of connections, PVCs and SVCs, may be used. Using PVCs simplifies the support of the IntServ model as the connections can then be considered as being simple point-to-point circuits and there is no difference from when RSVP is used over other leased line configurations. In contrast, if SVCs are used the problems of complexity and efficiency of setting up SVCs must be considered again as the use of RSVP complicates the situation. Basically, the problem is in deciding how many ATM connections must be set up to support the RSVP flows. Various options exist, ranging from using a single connection for all flows (which would be similar to the leased line solutions) to using different flow for each RSVP flow.[18]

Depending on the solution chosen, it decides where to concentrate the queuing and scheduling function of the edge router. If a single VC is established for all RSVP flows, the scheduling and queuing functionality must be done on the IP layer at the edge router so as to satisfy the QoS requirements. In contrast, if multiple VCs are established, the problem is simplified for the IP layer as it only needs to map the packets to the appropriate VCs. From this point on the ATM layering, queuing, and scheduling mechanisms would have to ensure the QoS guarantees.

Note that these problems of VC to QoS class mapping are complicated when the use of multicast connections is also considered. This is because RSVP basically supports heterogeneous receivers—receivers that differ in the QoS

18. While this same problem exists for the PVC case as well, complexity is more for the SVC case due to the increased flexibility.

requirements. This again means that there are a number of options in setting up the VCs for the relevant QoS classes. One method is to simply set up a different VC for each receiver. While this solution is simple to consider, it would be prohibitively expensive to implement. Another method is to set up a single multicast VC of a single QoS for all receivers. In this case the single QoS must be equal to the best QoS requested. Yet a third method would be to set up multiple multicast VCs for each different level of QoS such that each receiver could attach to the appropriate VC tree depending on its QoS level.

As mentioned in Section 7.1.1, in classical IPoA, the connection is normally released if the VC is idle for a certain amount of time. While this behavior is suitable for best-effort traffic, when the connection is established by RSVP, it must be assumed that the connection has been set up to offer a certain level of services. Consequently, the connection should not be released by the idle timeout function. The connection must be under the control of the RSVP state machine only.

Another major problem is how to map the IntServ and RSVP QoS parameters to the ATM connections. The IntServ QoS parameters are defined in the sender TSPEC, receiver TSPEC, and the receiver's RSPEC (Refer to Section 3.4). These parameters must be mapped to the ATM QoS parameters (PCR, SCR, and MBS). Details on how these mappings should be done for the three IntServ QoS classes are given in [23].

There are of course many more ATM traffic parameters than those mentioned above. Several ATM QoS parameters have no equivalent in the IntServ model's parameters, for example, the CLR, CDV, and CTD parameters. These must be set at the edge routers at the ingress point of the ATM network based on the configured parameters or other network operator–specific methods.

Guaranteed service and ATM QoS classes. The CBR and rt-VBR classes may both be used to support the guaranteed service class. While CBR is an obvious choice, CBR will lead to an inefficient use of bandwidth, as CBR will use up a certain amount of bandwidth whether or not traffic is present. When CBR is used to support guaranteed service, the PCR of the connection must be set to the peak rate of the guaranteed service.

In contrast to the CBR class, rt-VBR is a better match to supporting the bursty nature of Internet traffic. When rt-VBR is used to support guaranteed service, it is suggested that the SCR, PCR, and MBS of the ATM connection be related to the QoS parameters of the IntServ model [23, 25].

Controlled-load service and ATM QoS classes. The nrt-VBR and ABR may both be used to support the controlled-load service class. The UBR class by

itself is not appropriate for supporting the controlled-load service model, as it is very susceptible to network congestion. The CBR or rt-VBR classes are also not appropriate, as they will result in an inefficient use of network resources when used to support the bursty data traffic for which controlled-load service was defined. When nrt-VBR is used the PCR, SCR, and MBS are also related to the QoS parameters of the Intserv model [23].

When ABR is used the problem is simplified as the basic ABR service class offers similar service to that of the controlled-load service. The only parameter that needs to be set is the MCR of the ABR connection, which should be set to the minimum rate needed by the data source.

While UBR may be unacceptable to use as a basis for controlled load service, it may be possible to support controlled-load service under the newly defined differentiated UBR service type. As the service offered and guaranteed to a differentiated UBR connection is completely network-defined, this possibility may be a viable solution in the future.

Best-effort service and ATM QoS classes. Both UBR and ABR may be used to support the best-effort QoS class. The UBR class is an obvious choice for mapping best-effort IP traffic to ATM traffic classes. However, as explained in Section 7.5, naively mapping best-effort traffic to UBR connections can lead to a serious drop in performance as perceived by the user and aggravate any network congestion. Therefore, the network should implement some sort of frame discard method.

Using the ABR class is another way to support the best-effort QoS class. This would result in an efficient use of the ATM network resources while at the same time ensuring minimal cell loss for the best-effort packets. However, as was pointed out in Section 7.5, when ABR is used to support best-effort traffic in the overlay model, the performance of TCP/IP connections can drop seriously unless some intelligent queuing schemes are used in the edge routers.

7.6.1.2 Overlay Model and DiffServ over ATM (UNI 4.0)

It is also possible to support DiffServ QoS models over ATM connections. As with the IntServ case a number of mappings are possible.[19] The most obvious mapping is the use of CBR or rt-VBR connections to support the EF PHB. This is because the EF PHB tries to offer a "virtual circuit" like service

19. Note that unlike the IntServ model, there is no working group in the IETF working on the problem of mapping DiffServ QoS models and parameters to specific link layers. Consequently, this section is based on the authors' opinions.

with guaranteed bandwidth and minimal queuing delay for the packets that use this PHB. This is an obvious choice of mapping to the CBR QoS class. The use of the rt-VBR would also satisfy this criterion. The AF PHB may be supported in a number of different ways. Both nrt-VBR and ABR connections would be able to support the basic semantics of AF PHB. A crucial item in supporting the AF PHB is to ensure that the packet drop priorities are mapped to the appropriate CLP values. While the AF PHB defines three different loss priorities, the ATM cell is only able to carry a single CLP bit that can only indicate two cell loss priorities. Consequently, when mapping the AF PHB to the CLP bit either the top two or bottom two loss priorities must be tied together and marked with the same CLP bit. Note that it would be inappropriate to try to use different VCs for the different loss priorities in a single AF class as this may lead to out-of-sequence delivery at the receiver. This does not rule out different VCs for each AF class.

7.6.1.3 Overlay Model and DiffServ over ATM Using Differentiated UBR

As explained in Section 4.6.1, differentiated UBR is one of the new methods developed recently by the ATM Forum to more efficiently support TCP/IP traffic.[20] This is done by defining a new attribute to be associated with a UBR connection. Normally a UBR connection does not offer any service guarantees, basically offering a best-effort service. However, when differentiated UBR is used, a new attribute, the *behavior class*, may be associated with the UBR connection. The network may offer differentiated services to the user based on the behavior class associated with the user's connection. What the specific behavior classes are and whether the resources are specifically allocated are not specified. These are all up to the network operator's discretion to define and are employed as needed.

A BCS parameter is used to indicate the behavior class for the connection when the connection is initially set up. The values that the BCS parameter may take and the mapping with the actual behavior class in the network are not defined and may differ from network to network. The capability is applicable to all types of ATM connections including VPs, VCs, point-to-point, and point-to-multipoint connections.

It is easy to see that the definition and the specified functionality of differentiated UBR are extremely similar to that of the differentiated service

20. The actual specifications show two possible uses for the differentiated UBR mechanism. One is the support of differentiated services QoS model of the IETF. The other is the support of IEEE 802.1D user priorities that are used to provide service differentiation at the MAC layer.

model defined by the IETF. We can illustrate how this may be employed in an overlay network based on Figure 7.25. To support DiffServ in this case, multiple differentiated UBR ATM VC connections must be set up between each router. The edge routers (and also the interLIS routers) must also maintain a DSCP to VPI/VCI mapping table for each port. When a packet arrives at the router, the packet is classified based on its DSCP value to a certain PHB. Along with the "next-hop router" information from the router's routing table, this information is used to select the appropriate output port and the corresponding differentiated UBR ATM connection. Cell loss priority information is also passed to the ATM layer and encoded into the CLP bit of the ATM cell. The IP level QoS is guaranteed at the edge routers by the use of various queuing techniques. These QoS levels are maintained in the ATM level by using queuing and other traffic management techniques such as UPC and per-VC queuing.

7.6.2 Integrated Model and QoS Support

The integrated model currently consists of the MPLS-based solution for supporting IP traffic over ATM. As shown before, this model basically removes the whole ATM control stack and replaces it with an IP/MPLS control stack. As such, none of the predefined ATM methods for supporting QoS by using signaling and resource reservation are available. Instead, the MPLS control stack and the basic functions of the underlying ATM hardware must be used to offer this functionality.

7.6.2.1 Integrated Model and IntServ over MPLS/LC-ATM Switches

The key idea in supporting IntServ with MPLS/*label switch-controlled ATM* (LC-ATM) switches is to use RSVP for both label distribution/*label switch path* (LSP) setup and also for resource reservation. As resource reservation is the original function of RSVP only a method for label distribution with RSVP need to be added. This can be easily accomplished by defining a new object, the LABEL object, in all RESV messages [26]. Figure 7.26 shows

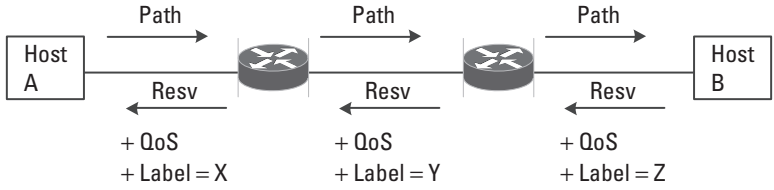

Figure 7.26 Label distribution and QoS reservation by RSVP messages [26].

how the labels may be distributed in the RESV messages. An LSR receiving an RESV message containing a LABEL object would update its *label-forwarding information base* (LFIB) with this label used as the outgoing label. The LSR should then allocate a new label to be used by the upstream node and include it in the RESV message and transmit it upstream. At the same time it should update the LFIB with this label. RSVP is shown here to support IntServ and label path setup. By the same mechanisms RSVP can also be used to solve traffic engineering problems when it is used to set up explicit paths.

7.6.2.2 Integrated Model and DiffServ over MPLS/LC-ATM Switches

To support DiffServ over MPLS there must be a method to map DiffServ DSCP values to the appropriate labels. This can be supported in two ways: one is so-called the *EXP bit-inferred PSC LSP* (E-LSP) method and the other is the *label-inferred PSC LSP* (L-LSP) method.[21] Figure 7.27 shows the difference between these two methods. Basically, E-LSP uses only one LSP and uses a field in the MPLS header to carry the DSCP code information, whereas L-LSP simply sets up a different LSP for each DSCP defined class.

In E-LSP, the three-bit EXP field in the MPLS label header is used. The packet is routed based on its label value, but at the output queue the queuing and scheduling behavior is decided by the EXP value. The EXP field

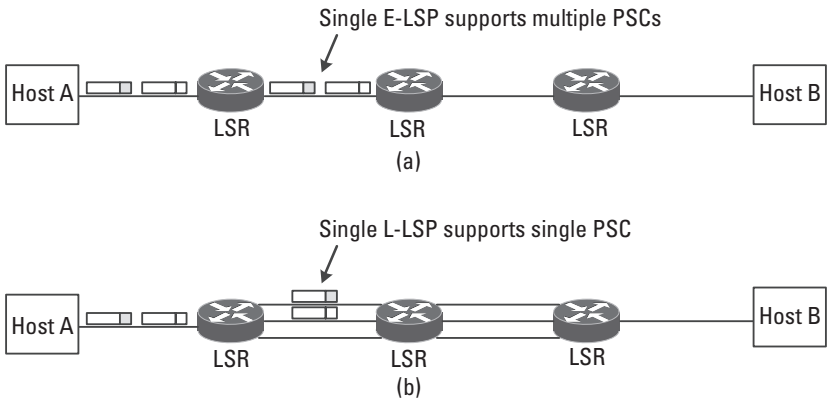

Figure 7.27 Ways of supporting DiffServ in MPLS networks: (a) E-LSP, and (b) L-LSP.

21. PSC is defined as the PHB scheduling class.

is only a 3-bit field, so the mapping of the 6-bit of DSCP to this field may result in loss of some information. The MPLS switches must be configured to map the EXP bits to the appropriate PHB behavior. However, for the case where MPLS-ATM switches are used, E-LSPs cannot be supported over LC-ATM interfaces because the ATM cells cannot carry the EXP bits in the label headers. Thus, only the L-LSP method may be used.

The L-LSP method maps each PHB to a different label or *forwarding equivalence class* (FEC). When DiffServ is run over MPLS-ATM, any number of L-LSPs per FEC may be allowed within a single MPLS ATM DiffServ domain. The basic compliance requirement is that the forwarding behavior experienced by a behavior aggregate forwarded over an L-LSP by the ATM LSR must be compliant with the DiffServ PHB specifications.

As only one CLP bit is available for encoding the drop priority, three PHB drop preference levels must be mapped to two levels. All ATM-LSRs must implement a frame discard mechanism such as EPD or PPD for performance improvements.

References

[1] Laubach, M., "Classical IP and ARP Over ATM," *RFC 1577*, January 1994.

[2] Bradley, T., and C. Brown, "Inverse Address Resolution Protocol," *RFC 1293*, Wellfleet Communications, Inc., January 1992.

[3] Atkinson, R., "Default IP MTU Over ATM AAL5," *RFC 1626*, Naval Research Laboratory, May 1994.

[4] Mogul, J. C., and S. E. Deering, "Path MTU Discovery," *RFC 1191*, November 1990.

[5] Luciani, J., et al., "NBMA Next Hop Resolution Protocol (NHRP)," *RFC 2332*, April 1998.

[6] IEEE, "IEEE Standards for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges," *IEEE 802.1D/p*, 1990.

[7] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridge Local Area Networks," *IEEE 802.1Q*, 1998.

[8] Przygienda, T., D. P. Siara, and R. Haas, "OSPF Over ATM and Proxy-PAR," *RFC 2844*, May 2000.

[9] Davidson, M., "ILMI-Based Server Discovery for ATMARP," *RFC2601*, June 1999.

[10] Davidson, M., "ILMI-Based Server Discovery for MARS," *RFC2602*, June 1999.

[11] Davidson, M., "ILMI-Based Server Discovery for NHRP," *RFC2603*, June 1999.

[12] Postel, J., and J. K. Reynolds, "Standard for the Transmission of IP Datagrams Over IEEE 802 Networks," *RFC 1042*, February 1988.

[13] Jacobson, V., "Compressing TCP/IP Headers for Low-Speed Serial Links," *RFC 1144*, February 1990.

[14] ATM Forum, "ATM User-Network Interface Specification, Version 3.1," Upper Saddle River, NJ: Prentice Hall, 1995.

[15] ATM Forum, "ATM User-Network Interface (UNI) Signaling Specification, Version 4.0," July 1996. Available at ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.000.ps.

[16] Braden, R., D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," *RFC 1633*, June 1994.

[17] Romanow, A., and S. Floyd, "Dynamics of TCP Traffic Over ATM Networks," *IEEE Journal on Selected Areas in Communications,* Vol. 13, No. 4, May 1995, pp. 633–641.

[18] Hassan, M., and M. Atiquzzaman, *Performance of TCP/IP over ATM Networks*, Norwood, MA: Artech House, 2000.

[19] ATM Forum Technical Committee, "Traffic Management Specification v4.1," af-tm-0056.000, April 1996.

[20] ATM Forum, "Addendum to Traffic Management Version 4.1 for an Optional Minimum Desired Cell Rate Indication for UBR," at-tm-0150.000, July 2000.

[21] Jagannath, S., and N. Yin, "End-to-End TCP Performance in IP/ATM Internetworks," ATM Forum Contribution 96-1711, December 1996.

[22] Kim, W. J., and B. G. Lee, "On Supporting TCP Traffic Over ABR Connections," *Proceedings of ICC' 98,* June 1998.

[23] Borden, M., and M. Garrett, "Interoperation of Controlled-Load and Guaranteed Service With ATM," *RFC 2381*, August 1998.

[24] ATM Forum, "Addendum to TM4.1: Differentiated UBR," at-tm-00149.000, July 2000.

[25] Ferguson, P., and G. Huston, *Quality of Service,* New York: John Wiley & Sons, 1998.

[26] Awduche, D., et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," *RFC 3209*, December 2001

Selected Bibliography

Andersen, N. E., et al., "Applying QoS Control Through Integration of IP and ATM," *IEEE Communications Magazine,* Vol. 38, No. 7, July 2000, pp. 130–136.

Armitage, G., "Support for Multicast Over UNI 3.0/3.1–Based ATM Networks," *RFC 2022*, November 1996.

ATM Forum, "ATM User-Network Interface Specification, Version 3.0," Englewood Cliffs, NJ: Prentice Hall, 1993.

ATM Forum, "ATM Traffic Management Specification, Version 4.0," April 1996. Available at ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps.

ATM Forum, "MPOA Baseline Version 1," May 1997.

ATM Forum, "PNNI Augmented Routing (PAR) Version 1.0," AF-RA-0104.000, January 1999.

ATM Forum Technical Committee, "LAN Emulation Over ATM, Version 1.0 Specification, af-lane-0021.000," January 1995.

ATM Forum Technical Committee, "LAN Emulation Over ATM Version 2—LUNI Specification," December 1996.

ATM Forum Technical Committee, "Traffic Management Specification v4.1," af-tm-0121.000, March 1999.

ATM Forum Technical Committee, "Private Network-Network Interface Specification v1.0 (PNNI)", March 1996.

Azcorra, A., et al., "IP/ATM Integrated Services Over Broadband Access Copper Technologies," *IEEE Communications Magazine,* Vol. 37, No. 5, May 1999, pp. 90–97.

Berger, L., "RSVP Over ATM Implementation Guidelines," *RFC 2379*, August 1998.

Berger, L., "RSVP Over ATM Implementation Requirements," *RFC 2380*, August 1998.

Borden, M., et al., "Integration of Real-Time Services in an IP-ATM Network Architecture," *RFC 1821*, August 1995.

Braden, R., et al., "Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification," *RFC 2205*, September 1997.

Broadband Integrated Service Digital Network (B-ISDN), "Digital Subscriber Signaling System No.2 (DSS2) User Network Interface Layer 3 Specification for Basic Call/Connection Control," ITU-T Recommendation Q.2931, (International Telecommunication Union: Geneva, 1994).

Cocca, R., M. Listanti, and S. Salsano, "Interaction of RSVP with ATM for the Support of Shortcut QoS Virtual Channels," *Proceedings of 2nd International Conference on ATM,* June 1999.

Crawley, E., et al., "A Framework for Integrated Services and RSVP Over ATM," *RFC 2382*, August 1998.

Eichler, G., et al., "Implementing Integrated and Differentiated Services for the Internet with ATM Networks: A Practical Approach," *IEEE Communications Magazine,* Vol. 38, No. 1, January 2000, pp. 132–141.

Floyd, S., and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking,* Vol. 3, No. 4, August 1995, pp. 365–386.

Garrett, M. W., "A Service Architecture for ATM: From Applications to Scheduling," *IEEE Network Magazine*, Vol. 10, No. 3, May 1996, pp. 6–14.

Georgatsos, P., et al., "Technology Interoperation in ATM Networks: The REFORM System," *IEEE Communications Magazine,* Vol. 37, No. 5, May 1999, pp. 112–118.

Heinanen, J., "Multiprotocol Encapsulation Over ATM Adaptation Layer 5," *RFC 1483*, July 1993.

Hong, D. P., and T. Suda, "Performance of ATM Available Bit Rate for Bursty TCP Sources and Interfering Traffic," *Computer Networks,* January 1999.

ITU-T Rec. I.311 (08/96) "B-ISDN General Network Aspects."

ITU-T Rec. I.311 Amendment 1 (03/2000) "B-ISDN General Network Aspects."

ITU-T Rec. I.321 (04/91) "B-ISDN Protocol Reference Model and Its Application."

ITU-T Rec. I.356 (10/96) "B-ISDN ATM Layer Cell Transfer Performance."

ITU-T Rec. I.361 (02/99) "B-ISDN ATM Layer Specification."

ITU-T Rec. I.362 (03/93) "B-ISDN ATM Adaptation Layer (AAL) Functional Description."

ITU-T Rec. I.363 (03/93) "B-ISDN ATM Adaptation Layer (AAL) Specification."

ITU-T Rec. I.363.1 (08/96) "Type 1 AAL."

ITU-T Rec. I.363.5 (08/96) "Type 5 AAL."

ITU-T Rec. I.371 (03/2000) "Traffic Control and Congestion Control in B-ISDN."

ITU-T Rec. I.371.1 (11/2000) "Guaranteed Frame Rate ATM Transfer Capability."

ITU-T Rec. I.381 (03/2001) "ATM Adaptation Layer (AAL) Performance."

Laubach, M., "Classical IP and ARP Over ATM," *RFC 2225*, April 1998.

Maher, M., "ATM Signaling Support for IP Over ATM—UNI Signaling 4.0 Update," *RFC 2331*, April 1998.

Mir, N. F., "An Efficient Multicast Approach in an ATM Switching Network for Multimedia Applications," *Journal of Network and Computer Applications,* Vol. 21, January 1998, pp. 31–39.

Mountzouris, I., et al., "Evaluation of the TCP Traffic Over the ABR Service Targeted To Support Mass Storage Applications," *Proceedings of ICATM*, 1999, pp. 85–90.

Orphanos, G., et al., "Compensating for Moderate Effective Throughput at the Desktop," *IEEE Communications Magazine,* Vol. 38, No. 4, April 2000, pp. 128–135.

Parekh, A. K., and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case," *IEEE/ACM Transactions on Networking,* Vol. 2, No. 2, April 1994, pp. 137–150.

Partridge, C., "A Proposed Flow Specification," *RFC 1363*, BBN, September 1992.

Pazos, C. M., M. R. Kotelba, and A. G. Malis, "Real-Time Multimedia Over ATM: RMOA," *IEEE Communications Magazine,* Vol. 38, No. 4, April 2000 pp. 82–87.

Perez, M., et al., "ATM Signaling Support for IP Over ATM," *RFC 1755*, February 1995.

Rajagopalan, B., et al., "A Framework for QoS-Based Routing in the Internet," *RFC 2386*, August 1998.

Shenker, S., C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," *RFC 2212*, September 1997.

Shenker, S., and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements," *RFC 2215*, September 1997.

Shiomoto, K., et al., "Scalable Multi-QoS IP+ATM Swith Router Architecture," *IEEE Communications Magazine,* Vol. 38, No. 12, December 2000, pp. 86–92.

Siara, D. P., and T. Przygienda, "OSPF Over ATM and Proxy-PAR," *RFC 2843*, May 2000.

Wroclawski, J., "Specification of the Controlled-Load Network Element Service," *RFC 2211*, September 1997.

Yashiro, Z., T. Tanaka, and Y. Doi, "Flexible ATM Switching Architecture for Multimedia Communications," *Proceedings of IEEE BSS'97*, 1997, pp. 58–64.